

**PENERAPAN ALGORITMA *SPEED UP ROBUST FEATURES*
(SURF) DAN *FEATURES FROM ACCELERATED
SEGMENT TEST (FAST)* UNTUK
PENGENALAN TEMPAT**

SKRIPSI

MEY HENDRA PUTRA SIRAIT

0701163068



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATRA UTARA
MEDAN
2020**

**PENERAPAN ALGORITMA *SPEED UP ROBUST FEATURS*
(SURF) DAN *FEATURES FROM ACCELERATED
SEGMENT TEST (FAST)* UNTUK
PENGENALAN TEMPAT**

SKRIPSI

Diajukan untuk Memenuhi Syarat Mencapai Gelar Sejarna Komputer

MEY HENDRA PUTRA SIRAIT

0701163068



**PROGRAM STUDI ILMU KOMPUTER
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUMATRA UTARA
MEDAN
2020**

PERSETUJUAN SKRIPSI

Hal : Surat Persetujuan Skripsi

Lamp : -

Kepada Yth.,

Dekan Fakultas Sains dan Teknologi

Universitas Islam Negeri Sumatera Utara Medan

Assalamu'alaikum Wr. Wb.

Setelah membaca, meneliti, memberikan petunjuk, dan mengoreksi serta mengadakan perbaikan, maka kami selaku pembimbing berpendapat bahwa skripsi saudara,

Nama	: Mey Hendra Putra Sirait
Nomor Induk Mahasiswa	: 0701163068
Program Studi	: Ilmu Komputer
Judul	: Penerapan Algoritma <i>Speed Up Robust Featurs</i> (SURF) Dan <i>Features From Accelerated Segment Test</i> (FAST) Untuk Pengenalan Tempat.

Dapat disetujui untuk segera di *munaqasyahkan*. Atas perhatiannya kami ucapkan terimakasih.

Medan, 29 Juli 2020M
08 Dzulhijah 1442H

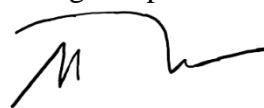
Komisi Pembimbing

Pembimbing Skripsi I



Dr. Mhd. Furqan, S.Si, M.Comp.Sc
NIP. 198008062006041003

Pembimbing Skripsi II



Rakhmat Kurniawan R, S.T., M.Kom
NIP. 198503162015031003

SURAT PERNYATAAN KEASLIAN SKRIPSI

Saya yang bertanda tangan dibawah ini:

Nama : Mey Hendra Putra Sirait
Nomor Induk Mahasiswa : 0701163068
Program Studi : Ilmu Komputer
Judul : Penerapan Algoritma *Speed Up Robust Featurs* (SURF) Dan *Features From Accelerated Segment Test* (FAST) Untuk Pengenalan Tempat

Menyatakan bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing disebutkan sumbernya. Apabila di kemudian hari ditemukan plagiat dalam skripsi ini maka saya bersedia menerima sanksi pencabutan gelar Akademik yang saya peroleh dan sanksi lainnya sesuai dengan peraturan yang berlaku.

Medan, 29 Juli 2020



Mey Hendra Putra Sirait
NIM. 0701163068

ABSTRAK

Struktur dari tempat yang sering kita jumpai berupa bangunan maupun jalan dengan seiring waktu berjalan, Hal itu menyebabkan kita terkadang kesulitan mengenal tempat tersebut dengan perubahan yang telah terjadi pada bangunan atau jalan. Teknologi saat ini berkembang dapat melakukan pencocokan antara sebuah citra dengan citra lain. Dalam mengenali sebuah citra maka perlu ada proses yang akan di lakukan dalam pencocokan citra namun pencocokan citra saat ini masih membandingkan piksel antar dua citra. Untuk membandingkan antara citra maka warna, resolusi dan bentuk dari piksel citra tersebut berpengaruh pada hasil pengenalan dalam citra. Untuk menangani masalah tersebut, algoritma yang dapat digunakan dalam proses kerja program ini adalah algoritma *Speed Up Robust Featurs* (SURF) dan *Features from Accelerated Segment Test* (FAST). FAST merupakan metode untuk menentukan sudut yang ada pada citra sedangkan algoritma SURF dapat mendeskripsi fitur yang ada pada sebuah citra sehingga pencocokan citra tidak lagi mencocokkan antar piksel tetapi berdsarkan pada deskriptor yang telah di hasilkan serta hasil yang dicocokkan akan tertera pada database, dengan menggunakan algoritma SURF, maka tidak perlu mengkhawatirkan resolusi, warna, dan bentuk dari citra yang akan dicocokkan. Pengujian yang dilakukan menghasilkan nilai *precision* 0,9 yang berarti nilai keberhasilan pencocokan adalah 9% dan dengan nilai *recall* mencapai 100% dan nilai yang sudah mencapai sampai 100% berarti jumlah titik mirip dengan jumlah titik yang sudah di cocokkan.

Kata Kunci: SURF, FAST, Pengenalan Citra, *Precision*, *Recall*

ABSTRACT

The structure of places that we often encounter in the form of buildings or roads over time. This causes us sometimes to find it difficult to recognize the place with changes that have occurred in buildings or roads. Currently developing technology can make a match between an image with another image. In recognizing an image, there needs to be a process that will be carried out in matching the image, but the current image matching is still comparing the pixels between the two images. To compare between images, the color, resolution and shape of the image pixels affect the recognition results in the image. To deal with this problem, the algorithms that can be used in the work process of this program are the Speed Up Robust Features (SURF) and Features from Accelerated Segment Test (FAST) algorithm. FAST is a method for determining the angle that is in an image while the SURF algorithm can describe the features that exist in an image so that image matching no longer matches between pixels but based on the descriptors that have been generated and the matched results will be listed in the database, using the SURF algorithm. , there is no need to worry about the resolution, color, and shape of the image to be matched. Tests conducted produce a precision value of 0.9, which means that the value of successful matching is 9% and with a recall value of 100% and a value that has reached 100%, it means that the number of points is similar to the number of points that has been validated.

Keywords: SURF, FAST, Image Recognition, Precision, Recall

KATA PENGANTAR



Assalamu'alaikum Warahmatullahi Wabarakatuh.

Saya selaku penulis mengucapkan Alhamdulillah dan puji syukur atas kehadiran Allah SWT, yang telah melimpahkan Rahmat dan Karunia-Nya, sehingga penulis dapat merampungkan skripsi dengan judul: Penerapan Algoritma *Speed Up Robust Features* (SURF) dan *Features Form Accelerated Segment Test* (FAST) untuk pengenalan tempat. Ini untuk memenuhi salah satu syarat menyelesaikan studi. Tak lupa juga sholawat serta salam semoga senantiasa tercurahkan kepada uswatun hasanah kita yaitu, Nabi Muhammad SAW beserta keluarga, para sahabat, dan pengikutnya termasuk kita semua yang senantiasa menantikan syafa'atnya kelak di hari akhir.

Penghargaan dan terima kasih yang setulus-tulusnya kepada Ayahanda tercinta Alm. Ramlan Sirait Bin Jafar Sirait dan Ibunda yang kusayangi Ramsiah Siregar yang telah mencurahkan segenap cinta dan kasih sayang serta perhatian moril maupun materil. Semoga Allah SWT selalu melimpahkan Rahmat Kesehatan, Karunia, serta keberkahan di dunia dan di akhirat atas segala budi baik yang telah diberikan kepada penulis. Penghargaan dan terima kasih penulis berikan kepada bapak Dr. Mhd. Furqan, S.Si., M.Comp.Sc. selaku Pembimbing I dan bapak Rakhmat Kurniawan R, S.T., M.Kom selaku Pembimbing II yang telah membantu penulisan skripsi ini. Serta ucapan terima kasih kepada:

- 1 Bapak Prof. Dr. Saidurrahman, M.Ag, selaku Rektor Universitas Islam Negeri Sumatera Utara Medan
- 2 Bapak Dr. H. M. Jamil, MA, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan
- 3 Bapak Dr. Mhd Furqan, S,Si., M.Comp.Sc selaku Ketua Program Studi Ilmu Komputer dan selaku dosen pembimbing skripsi I yang telah berkontribusi membantu penulis dalam memberikan ide, saran, keritikan, dan

bimbingannya kepada penulis selama penulis mengerjakan proposal skripsi ini.

- 4 Bapak Ilka Zufria, M.Kom selaku Sekretaris Program Studi Ilmu Komputer.
- 5 Bapak Rakhmat Kurniawan R, S.T., M.Kom selaku dosen pembimbing akademik dan selaku dosen pembimbing skripsi II yang telah berkontribusi membantu penulis dalam memberikan ide, saran, keritikan, dan bimbingannya kepada penulis selama penulis mengerjakan proposal skripsi ini.
- 6 Seluruh tenaga pengajar dan pegawai program studi S1 Ilmu Komputer Fakultas Sains dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
- 7 Teman-teman kelas Ilmu Komputer 2 yang selalu memberikan dukungan serta arahan kepada penulis.
- 8 Kepada Sahabat-Sahabat penulis yang selalu memberikan *support* dan memberikan masukan kepada penulis.
- 9 Dan semua pihak yang telah membantu penulis namun tidak dapat disebutkan satu persatu.

Akhir kata penulis menyadari bahwa dalam penulisan skripsi ini masih jauh dari kesempurnaan. Karena itu, penulis memohon saran dan kritik yang sifatnya membangun demi kesempurnaannya dan semoga bermanfaat bagi kita semua. AamiinYa Rabbal'alam.

Medan, 18 Mei 2020

Hormat Saya



Mey Hendra Putra Sirait

DAFTAR ISI

	Hal
ABSTRAK	i
ABSTACT.....	ii
KATA PENGANTAR.....	iii
DAFTAR ISI.....	v
DAFTAR GAMBAR.....	vii
DAFTAR TABLE	ix
DAFTAR LAMPIRAN	x
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Tujuan Penelitian	3
1.4 Manfaat Penelitian	3
1.5 Batasan Masalah	3
BAB II TINJAUAN PUSTAKA	
2.1 Definisi Citra	4
2.2 Citra Digital	4
2.3 Citra Analog	5
2.4 Tipe Citra	5
2.5 Format File pada Citra	7
2.6 Pengolahan Citra	9
2.7 Ekualisasi Histogram	9
2.8 <i>Thresholding</i>	10
2.9 <i>Features From Accelerated Segment Test (FAST)</i>	13
2.10 <i>Speeded Up Robust Features (SURF)</i>	15
2.11 Ekstraksi Fitur	19

2.12 Sistem Koordinat Warna	20
2.13 <i>Precision Dan Recall</i>	20
2.14 Penelitian Terdahulu	20
2.15 Visual Studio, C#, XAMPP, dan Mysql.....	22

BAB 3 METODOLOGI PENELITIAN

3.1 Tempat dan Waktu Penelitian	24
3.1.1 Tempat Penelitian.....	24
3.1.2 Waktu dan Jadwal Pelaksanaan Penelitian	24
3.2 Bahan dan Alat Penelitian	25
3.2.1 Perangkat Keras	25
3.2.2 Perangkat Lunak	25
3.3 Proses Kerja	25
3.3.1 Proses Pencarian Tempat	25
3.3.2 Proses Tampilan Data	27
3.3.3 Proses Melihat Lokasi	28
3.3.4 Proses Pendeteksian sudut dengan metode FAST.....	28
3.3.5 Proses Fitur dengan metode SURF	29
3.3.6 Analisis Kebutuhan	29
3.3.7 Perancangan	30

BAB IV HASIL DAN PEMABAHASAN

4.1 Pembahasan	35
4.1.1 Analisis Data	35
4.1.2 Representasi Data.....	35
4.1.3 Hasil Analisis Data.....	48
4.1.4 Perancangan	48
4.2 Hasil	52
4.2.1 Pengujian.....	52
4.2.2 Penerapan	55

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan.....	56
---------------------	----

5.2 Saran.....	56
DAFTAR PUSTAKA	57

DAFTAR GAMBAR

Gambar	Judul Gambar	Halaman
2.1	Citra Biner Dengan Nilai Piksel 0 Atau 1	6
2.2	Citra Grayscale Dengan Nilai Piksel 0 Atau 255	6
2.3	Citra Warna Dengan Komponen Warna RGB	7
2.4	Fungsi $S=T(R)$ Dengan <i>Thresholding</i> M	11
2.5	Contoh <i>Thresholding</i> Pada Kuantisasi Citra	12
2.6	Histogram <i>Thresholding</i>	13
2.7	Deteksi sudut dengan algoritma FAST	15
2.8	Membangun <i>Descriptor</i> Dengan Metode SURF	17
3.1	Contoh Citra Pada Proses Pencocokan	26
3.2	<i>Flowchart</i> Pencarian Tempat	26
3.3	<i>Flowchart</i> Tampil Data	27
3.4	<i>Flowchart</i> Melihat Tempat	28
3.5	Tampilan Pada <i>Form</i> Pertama.....	30
3.6	<i>Form</i> Pertama Pada Menu.....	31
3.7	<i>Form</i> Untuk Proses	32
3.8	<i>Form</i> Pada Tampilan.....	33
3.9	<i>Form</i> Pada Pengujian	34
4.1	Potongan Citra Yang Di Ambil.....	35
4.2	<i>Form</i> Awal	48
4.3	<i>Form</i> Cari	49
4.4	<i>Menu</i> Tampil	49
4.5	<i>Form</i> Penguji.....	50
4.6	<i>Form</i> Tentang.....	51

4.7 <i>Form</i> Keluar	51
4.8 Penginputan Citra	52
4.9 Proses Pencarian Yang Di <i>Input</i>	53
4.10 Hasil Dari Pencocokan Citra.....	53
4.11 Tampilan Pencocokan Sudut Pada Citra	54
4.12 Tampilan Mirip Pada Citra.....	54
4.13 <i>Rute</i> Lokasi.....	55

DAFTAR TABLE

Table	Judul Table	Halaman
3.1	Waktu Pelaksanaan Penelitian	25
4.1	Nilai RGB Dari Potongan Citra	36
4.2	Nilai Citra Setelah Konveksi <i>Grayscale</i>	39
4.3	Nilai Citra Untuk Proses Histogram <i>Equalization</i>	40
4.4	Nilai Citra Setelah Proses Histogram	41
4.5	Deteksi Sudut Dengan Metode FAST	41
4.6	Pencarian Nilai Vektor Untuk <i>Descriptor</i>	44
4.7	Nilai Vektor Untuk <i>Descriptor</i>	45
4.8	<i>Descriptor</i> citra input pada titik a pada sistem koordinat warna	45
4.9	<i>Descriptor</i> citra database pada titik a untuk koordinat warna	46
4.10	Hasil <i>euclidean distance</i> pada sistem koordinat warna	46
4.11	Hasil pencocokan	46

DAFTAR LAMPIRAN

Lampiran	Judul Lampiran
1.	Listing Program
2.	Kartu Bimbingan Skripsi
3.	Daftar Riwayat hidup

BAB I PENDAHULUAN

1.1 Latar Belakang

Dalam ayat-ayat tersebut dijelaskan bahwa Allah mengajari (memberi) suatu ilmu kepada manusia yang tidak diberikannya kepada malaikat. Allah mengetahui segala yang terlahir maupun yang tersembunyi (di dalam hati) dan ilmu Allah sangat luas, meliputi segala rahasia yang ada dilangit dan di bumi. Ilmu yang diberikan Allah kepada manusia hanya sebagian kecil saja dari seluruh ilmu Allah, seperti yg tercermin dalam firman Allah



Artinya : “(1)Bacalah dengan (menyebut) nama Tuhanmu Yang Menciptakan.(2)Dia menciptakan manusia dari segumpal darah.(3)Bacalah dan Tuhanmulah Yang Maha Pemurah. (4)Yang Mengajar (manusia) dengan perantaraan kalam (tulisan baca).(5)Dia Mengajarkan manusia apa yang tidak diketahuinya.” (QS al-‘Alaq: 1-5)”. Ayat-ayat dalam Al-Qur’an selalu merangsang akal manusia untuk berpikir lebih lanjut tentang isi ayat-ayatnya yang banyak menyangkut tentang ilmu pengetahuan dan teknologi.

Dengan kemajuan suatu teknologi yang berkembang tidak dapat di pungkirkan bahwa teknologi membantu meringankan pekerjaan pada manusia. Begitu juga pada suatu tempat yang seiring berjalannya waktu dapat memungkinkan bahwa adanya terjadi perubahan struktur seperti jalan, bangunan-

bangunan atau tata letak pada bentuk bangunan, sehingga untuk mengenali tempat menjadi sulit dan pengenalan tempat sangat dibutuhkan sesuai dengan kebutuhan informasi mengenai tempat tersebut, untuk dapat mengenali tempat tersebut maka dibutuhkan suatu proses dalam pencocokan citra antara citra *input* dengan citra yang ada di *database*. Pencocokan dalam citra ini merupakan suatu proses pengambilan dua citra untuk dibandingkan antar piksel dengan tujuan memperoleh area fisik yang sama dari pencitraanya. Pencocokan citra bertujuan untuk mengetahui identitas citra yang dicocokkan. Permasalahan yang terjadi dalam pencocokan citra adalah perbedaan resolusi, ukuran, warna, posisi, jenis, pola dan bentuk dari dua buah citra yang akan dicocokkan.

Dalam proses kerja citra yang di proses antar 2 citra bertujuan untuk mengetahui tempat atau mengenali tempat tersebut yang di mana akan di sesuaikan di dalam *database*. Maka dari itu untuk menangani masalah tersebut maka algoritma yang dapat digunakan dalam proses kerja program ini adalah algoritma *Speeded Up Robust Features* (SURF) dan *Features from Accelerated Segment Test* (FAST). FAST merupakan algoritma untuk menentukan sudut yang ada pada citra sedangkan algoritma SURF dapat mendeskripsi fitur yang ada pada sebuah citra sehingga pencocokan citra tidak lagi mencocokkan antar piksel tetapi berdsarkan pada deskriptor yang telah di hasilkan.

Dalam penelitian ini pengenalan tempat dengan pencocokan citra dengan algoritma SURF yang di kombinasikan dengan FAST. Serta informasi yang dihasilkan berupa lokasi dari suatu tempat dan tingkat akurasi dari pengenalan tempat. Berdasarkan uraian sebelumnya, maka diajukan sebagai proposal skripsi denan judul “**Penerapan algoritma *Speed Up Robust Featurs* (SURF) dan *Features From Accelerated Segment Test* (FAST) untuk Pengenalan Tempat** “

1.2 Rumusan Masalah

Berdasarkan latar belakang di atas maka rumusan masalah sebagai berikut:

1. Bagaimana mengetahui suatu citra yang akan di cocokkan antara ke 2 citra dari warna, resulusi dan sudut pandang dalam pengambilan foto.

2. Bagaimana dalam pencocokan dua citra 2 dimensi yang akan memberikan kecocokan antara gambar yang di input dengan gambar yang terdapat pada database dengan dibutuhkannya algoritma *Speeded Up Robust Features* (SURF) yang dapat untuk melakukan menghasilkan suatu deskriptor dan algoritma *Features From Accelerated Segment Test* (FAST) yang dapat untuk mempercepat suatu pendeteksian.

1.3 Tujuan Penelitian

1. Menjelaskan pencocokan antara 2 gambar yang akan di proses untuk mendapatkan hasil yang akan di cocokan dalam database tersebut.
2. Tujuannya dapat disampaikan yaitu menghasilkan suatu aplikasi yang dapat mengenali suatu tempat dengan menggunakan citra 2 dimensi dengan menggunakan metode *Speeded Up Robust Feature* (SURF) dan algoritma *Features from Accelerated Segment Test* (FAST).

1.4 Manfaat Penelitian

Manfaat pada penelitian ini sebagai berikut :

1. Pengaplikasian yang dapat mampu untuk memahami atau mengenali suatu tempat yang menggunakan suatu pemasukan berupa sebuah citra.
2. Memahami cara suatu citra dalam mengenali suatu tempat dengan algoritma SURF dan juga dengan algoritma FAST.

1.5 Batasan Masalah

Adapun Batasan Masalah yang dilakukan yaitu :

1. Pada data awal yang terdapat pada database berupa citra tempat atau gedung yang terdapat pada sekitaran area kampus 1 Universitas Islam Negeri Sumatera Utara Medan.
2. Pencocokan pada citra hanya berlaku untuk 2 dimensi.
3. Untuk jenis file citra dibatasi dengan *jpg/jpeg* dan *png*

BAB II

TINJAUAN PUSTAKA

2.1 Definisi Citra

Sebuah Citra dapat didefinisikan sebagai sebuah fungsi dua dimensi. $f(x,y)$, dimana x dan y adalah koordinat bidang (spasial), dan amplitudo dari tiap pasangan koordinat (x,y) dinamakan dengan intensitas citra pada titik tersebut. Istilah level keabuan mengacu pada intensitas dari citra monokrom, Sebagai contoh, pada sistem warna RGB, sebuah citra memuat tiga citra komponen individual (*red*, *green*, dan *blue*). Karna itu Banyak Teknik yang di kembangkan untuk citra monokrom dapat di perluas untuk citra warna dengan memproses ketiga citra komponen secara terpisah. (Sianipar,RH, 2018)

Sebuah Citra dapat bernilai kontinyu terhadap koordinat (x,y) dan juga terhadap amplitudo. Konversi citra semacam itu ke dalam bentuk digital mengharuskan koordinat dan amplitudo digitalisasi. Pendigitalan nilai-nilai koordinat. Jadi ketika x,y , dan amplitudo bernilai diskrit anda dapat menamakan citra tersebut sebagai citra digital (Sianipar,RH, 2018).

2.2 Citra Digital

Citra adalah suatu representasi (gambaran), kemiripan, atau imitasi dari suatu objek. Citra terbagi 2 yaitu ada citra yang bersifat analog dan ada citra yang bersifat digital. Citra analog adalah citra yang bersifat kontinu seperti gambar pada monitor televisi, foto sinar X, hasil CT *Scan* dll. Sedangkan pada citra digital adalah citra yang dapat diolah oleh komputer. (T,Sutoyo et al. 2009: 9)

Citra digital merupakan representatif dari citra yang di ambil oleh mesin dengan bentuk pendekatan berdasarkan sampling dan kuantisasi. Sampling menyatakan besarnya kotak-kotak yang disusun dalam baris dan kolom. Dengan kata lain, sampling pada citra menyatakan besar kecilnya ukuran piksel (titik) pada citra, dan kuantisasi menyatakan besarnya nilai tingkat kecerahaan yang di

nyatakan dalam nilai tingkat keabuan (*grayscale*) sesuai dengan jumlah bit biner yang di gunakan oleh mesin, dengan kata lain kuantisasi pada citra menyatakan jumlah warna yang ada pada citra (Basuki, 2006).

2.3 Citra Analog

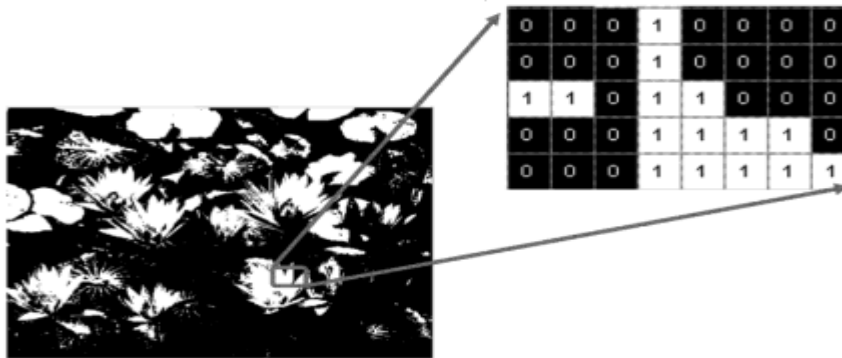
Citra Analog ada citra yang bersifat kontiniu. Contoh citra analog adalah citra pada monitor televisi, foto, sinar x, lukisan, pemandangan alam, hasil *CT scan*, citra yang terekam dalam pita kaset dan lain sebagainya. Citra analog tidak bisa dipresentasikan dalam komputer sehingga tidak dapat di proses pada komputer secara langsung. Agar dapat di proses di komputer, citra harus terlebih dahulu di konversi dari analog ke digital. Citra analog dihasilkan dari alat-alat analog, seperti video kamera analog, kamera foto analog, *CT scan*, *sensor ultrasound*, dan lain-lain (Sutoyo, et al, 2009).

2.4 Tipe Citra

Ada beberapa tipe citra yang sering digunakan untuk penelitian, di antaranya adalah :

1. Citra Biner

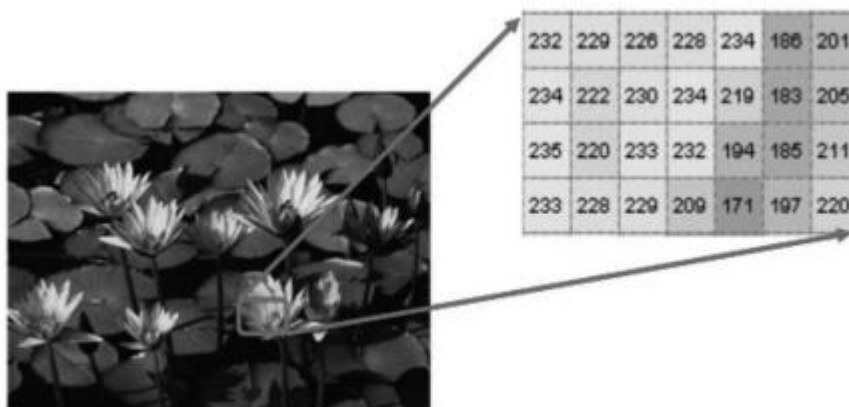
Didalam Citra Biner, tiap-tiap piksel hanya membutuhkan 1 bit memori. Maka dengan demikian, setiap piksel hanya mempunyai 2 (dua) buah kemungkinan nilai intensitas, yaitu 1 atau 0 . gambar 2.1 menunjukkan citra biner dilihat dari dekat dengan beberapa nilai intensitas piksel. (Andono Pulung Nurtantio, Sutojo.T, Muljono, 2017)



Gambar 2.1 Citra Biner dengan nilai piksel 0 atau 1 (Sumber : Andono Pulung Nurtantio, Sutojo.T, Muljono, 2017)

2. Citra *Grayscale*

Citra *Grayscale* adalah matriks data yang nilai-nilainya mewakili intensitas setiap piksel berkisar antara 0 sampai 255. Setiap piksel membutuhkan 8bit memori. Gambar 2.2 menunjukkan citra *grayscale* dilihat dari dekat dengan beberapa nilai intensitas piksel. (Andono Pulung Nurtantio, Sutojo.T, Muljono, 2017)

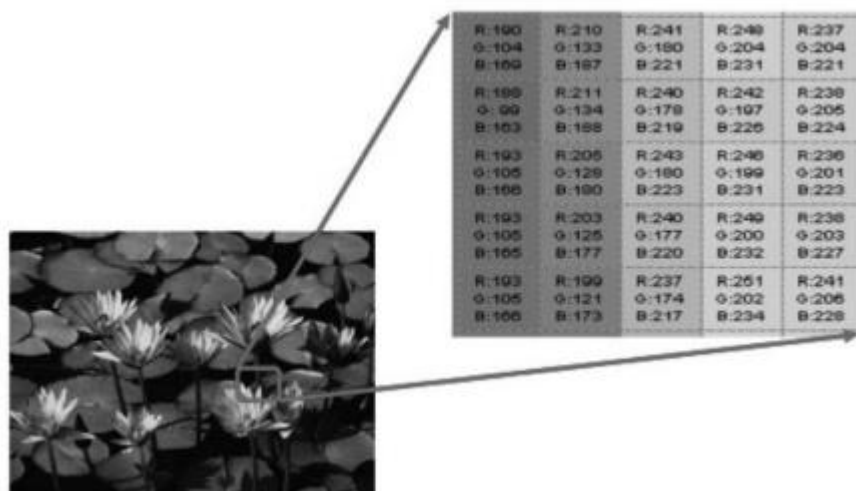


Gambar 2.2 Citra *Grayscale* dengan nilai piksel 0 atau 255 (Sumber : Andono Pulung Nurtantio, Sutojo.T, Muljono, 2017)

3. Citra Warna

Citra Warna adalah citra yang masing-masing piksel mempunyai 3 (tiga) komponen warna yang spesifik, yaitu komponen merah (*red*), hijau (*green*), dan biru (*blue*). Warna setiap piksel ditentukan oleh kombinasi dari intensitas warna

merah, hijau, dan biru yang disimpan pada bidang warna diokasi piksel. Format file grafis menyimpan citra warna sebagai citra 24 bit yang berasal dari komponen merah, hijau, dan biru masing-masing 8bit. Hal ini menyebabkan citra warna mempunyai 24 juta kemungkinan warna. Gambar 2.3 menunjukkan citra warna dilihat dari dekat dengan beberapa nilai intensitas piksel. (Andono Pulung Nurtantio, Sutojo.T, Muljono, 2017)



Gambar 2.3 Citra warna dengan komponen warna R(red), G(green), dan B (blue)
(Sumber : Andono Pulung Nurtantio, Sutojo.T, Muljono, 2017)

2.5 Format File pada Citra

Format File Citra standart yang digunakan saat ini terdiri dari beberapa jenis. Format-format ini digunakan dalam menyimpan citra dalam sebuah file. Setiap format memiliki karakteristik masing-masing. Berikut adalah penjelasan beberapa format umum digunakan saat ini ini : (Putra Darma,2010)

1. *Bitmap (*.bmp)*, Format *bmp* adalah format penyimpanan standard tanpa kompresi yang umum dapat digunakan untuk menyimpan citra biner hingga citra warna. Format ini terdiri dari beberapa jenis yang setiap jenisnya ditentukan dengan jumlah bit yang digunakan untuk menyimpan citra sebuah nilai piksel.

2. *JPEG(*.jpeg)*, Format *jpg* adalah format yang sangat umum digunakan saat ini, khususnya untuk transmisi citra. Format ini digunakan untuk menyimpan citra hasil kompresi dengan metode *JPEG*.
3. *Graphics Interchange Format (*.gif)*, Format ini dapat digunakan pada citra warna dengan palet 8 bit. Pada umumnya digunakan pada aplikasi *web*. Kualitas yang rendah menyebabkan format ini tidak terlalu populer di kalangan peneliti pengolahan citra digital.
4. *Tagged Image Format (*.tif/*.tiff)*, Format ini merupakan format citra yang dapat digunakan untuk menyimpan citra *bitmap* hingga citra dengan warna *palette* kompresi. Format ini dapat digunakan untuk menyimpan citra yang tidak terkompresi dan juga citra terkompresi.
5. *Portable Network Graphics (*.png)*, Format *png* adalah format penyimpanan citra terkompresi. Format ini dapat digunakan pada citra *grayscale*, citra dengan palet warna dan juga citra *full color*. Format *.png* juga mampu menyimpan informasi sehingga kanal *alpha* dengan penyimpanan sebesar 1 hingga 16 bit perkanal.
6. *MPEG (*.mpg)*, Format ini merupakan format penyimpanan citra yang diperuntukan sebagai format penyimpanan citra bergerak (video).
7. *RGB(*.rgb)*, Format ini merupakan format penyimpanan citra yang dibuat oleh *silicongraphics* untuk menyimpan citra berwarna.
8. *RAS (*.ras)*, Format ini digunakan untuk menyimpan citra RGB tanpa kompresi.
9. *Postscript (*.pas, *.epas, *.epfs)*, Format ini diperkenalkan sebagai format untuk menyimpan citra buku elektronik. Dalam format ini, citra direpresentasikan kedalam deret nilai *decimal* atau *heksa desimal* yang dikodekan kedalam ASCII.
10. *Portable Image File Format*, Format ini memiliki beberapa bagian di antaranya adalah diantaranya adalah *portable bit map*, *portable graymap*, *portable pixmap* dan *portable network map*, dengan format berturut-turut adalah *(.pbm)*, *(.ppm)*, dan *(.pnm)*.

2.6 Pengolahan Citra

Secara umum, pengolahan citra digital menunjuk pada pemrosesan gambar 2 dimensi menggunakan komputer. Dalam konteks yang lebih luas, pengolahan citra digital mengacu pada pemrosesan setiap data 2 dimensi. Citra digital merupakan sebuah larik (*array*) yang berisi nilai-nilai *real* maupun kompleks yang dipresentasikan dengan deretan bit tertentu. (Darma Putra, 2010)

Pengolahancitra digital adalah sebuah disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas citra (peningkatan kontras, transformasi warna, restorasi citra), transformasi citra (notasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (*feature images*) yang optimal untuk tujuan analisis, melakukan proses penarikan, informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. *Input* dari pengolahan citra adalah citra, sedangkan *outputnya* adalah citra hasil pengolahan. (Sutoyo, et al, 2009).

Pengolahan citra pada dasarnya mencakup tiga langkah berikut (Munir, 2004):

- 1 Mengimpor citra melalui alat akuisisi citra.
- 2 Menganalisis dan memanipulasi citra.
- 3 *Output* yang dihasilkan dapat diubah citra atau laporan yang didasarkan pada analisis citra.

Melalui proses transformasi, suatu citra dapat dinyatakan sebagai kombinasi linier dari sinyal dasar (*basis signals*) yang sering disebut dengan fungsi basis (*basis function*). Suatu citra yang telah mengalami transformasi dapat diperoleh kembali dengan menggunakan transformasi balik (*inver strans formation*). Tujuan diterapkannya transformasi citra adalah untuk memperoleh informasi (*Feature extraction*) yang lebih jelas yang terkandung dalam suatu citra. (Putra,2006).

2.7 Ekualisasi Histogram

Ekualisasi Histogram merupakan suatu cara yang bertujuan untuk memperoleh histogram dengan intensitas terdistribusi secara seragam pada citra. Pendekatan

yang dilakukan adalah untuk mendapatkan aras keabuan yang lebih luas pada daerah yang banyak memiliki banyak pixel dan mempersempit aras keabuan pada daerah berpixel sedikit. Efeknya dapat digunakan untuk meningkatkan kontras secara menyeluruh. Perlu diketahui proses perataan histogram, dimana nilai derajat keabuan pada suatu citra dibuat rata. Untuk dapat melakukan histogram ekualisasi ini diperlukan suatu fungsi distribusi komulatif yang merupakan komulatif dari histogram. Histogram ekualisasi (perataan histogram) adalah suatu proses dimana histogram diratakan berdasarkan fungsi linear (garis lurus). (Novidiantoko Dwi, 2018)

Bila suatu citra berukuran $M \times N$ akan dinormalisasi dengan HE pada range $[0, 255]$, maka rumus untuk HE adalah sebagai berikut (Hum, Y.C et al, 2004):

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1)\right) \dots \dots \dots (2.3)$$

Dimana:

$h(v)$ = target intensitas pada derajat keabuan v

$cdf(v)$ = *cumulative distribution function* / akumulasi jumlah piksel pada derajat keabuan v

cdf_{min} = cdf paling minimum pada skala keabuan pertama

L = 256 (intensitas warna keabuan)

M = lebar citra

N = tinggi citra

Bila citra ingin dinormalisasi ke *range* $[1, 255]$, maka rumus HE yang berlaku adalah sebagai berikut (Hum, Y.C et al, 2004):

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 2)\right) + 1 \dots \dots \dots (2.4)$$

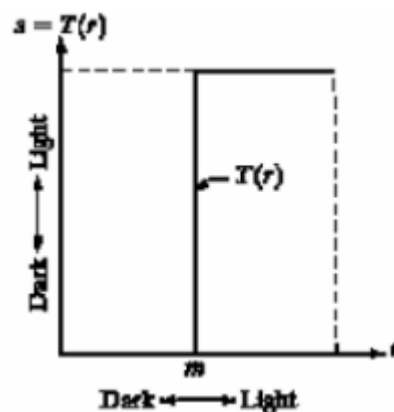
Bila hasil histogram akan dinormalisasi ke nilai tertentu atau *range* [*Batas Bawah*, *Batas Atas*] maka rumus HE yang berlaku adalah (Hum, Y.C et al, 2004):

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (BatasAtas - BatasBawah)\right) + BatasBawah \dots \dots \dots (2.5)$$

Setiap intensitas keabuan v diubah intensitas nilainya ke target intensitas $h(v)$, maka hasilnya adalah citra yang mendapat pemerataan histogram dari range Batas Bawah ke Batas Atas (Hum, Y.C et al, 2004).

2.8 Thresholding

Pemakaian *thresholding* dapat menghasilkan citra biner (dua level warna), dengan fungsi $T(r) = m$ sebagai nilai *thresholding*. Piksel dengan level warna $r \leq m$ akan berubah menjadi warna hitam atau $s=0$, sedangkan piksel dengan level warna $r \geq m$ berubah menjadi warna putih atau $s=255$, perhatikan gambar 2.4. fungsi transformasi untuk citra biner dengan *thresholding* m . (Ramadijanti, Nana dkk,2014)

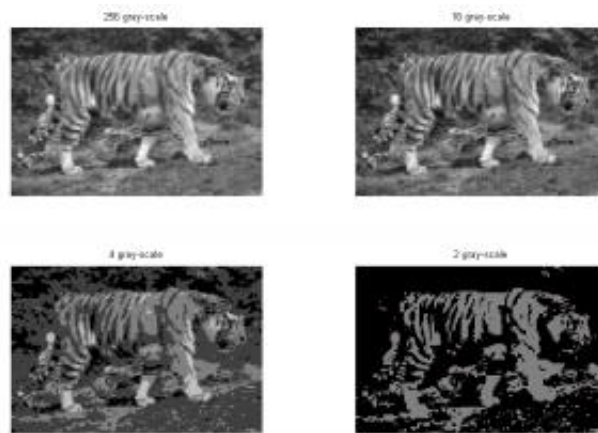


Gambar 2.4 Fungsi $s=T(r)$ dengan *Thresholding* m .

(Sumber Ramadijanti, Nana dkk,2014)

Thresholding juga dapat digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. Dengan menggunakan *thresholding* maka derajat keabuan bisa diubah sesuai keinginan, misalkan diinginkan menggunakan derajat keabuan 16, maka tinggal membagi nilai derajat keabuan dengan 16. Proses *thresholding* disini adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan a dapat digunakan rumus: $b = \text{int}(w/x)$ dimana : w adalah nilai derajat keabuan sebelum *thresholding* x adalah nilai derajat

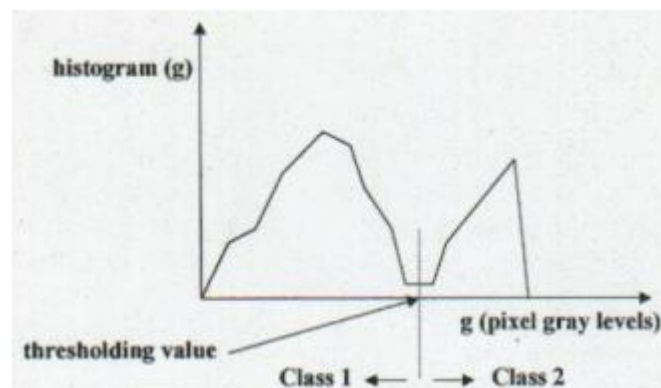
keabuan setelah *thresholding*) (a b int=256 Berikut ini contoh *thresholding* mulai di 256, 16, 4 dan 2 pada gambar 2.5. (Ramadijanti, Nana dkk,2014)



Gambar 2.5. Contoh *Thresholding* Pada Kuantisasi Citra .

(Sumber Ramadijanti, Nana dkk,2014)

Pada *thresholding* yang tinggi, hampir tidak tampak perbedaan karena keterbatasan mata, tetapi pada nilai *threshold* yang kecil seperti 2, 4, 8 dan 16 tampak sekali perbedaannya. Untuk *thresholding* 2, ini sama artinya dengan mengubah citra menjadi citra biner. Pemakaian *thresholding* yang lain dapat digunakan pada proses segmentasi citra dengan menggunakan histogram *thresholding*. Misalkan histogram gray - level dari citra, $f(x, y)$, yang terdiri dari obyek gelap dengan latar belakang terang, sedemikian sehingga piksel *grayscale* pada objek dan latar belakang di kelompokkan menjadi dua daerah dominan. Salah satu cara yang untuk mengekstrak objek dari latar belakang dengan memilih ambang batas 'T' yang memisahkan dua daerah. Maka 51 setiap piksel (x, y) dimana $f(x, y) > T$ disebut piksel obyek, jika tidak berarti piksel tersebut disebut piksel latar belakang. Contoh histogram *thresholding* pada gambar 2.6 .(Ramadijanti, Nana dkk,2014).



Gambar 2.6 Histogram *Thresholding* .

(Ramadijanti, Nana dkk,2014)

2.9 Features From Accelerated Segment Test (FAST)

Deteksi sudut digunakan sebagai tahapan pertama dalam berbagai pengolahan seperti *tracking*, SLAM (*simultaneous, localisation, and mapping*), *localization*, pencocokan citra dan pengenalan citra (Rosten & Drummond, 2006).

Karena pengujian tidak menghitung fungsi respons sudut, maka *non-maximal suppression* tidak dapat diterapkan langsung ke fitur yang dihasilkan. Akibatnya, V harus dihitung untuk setiap sudut yang terdeteksi, dan *non-maximal suppression* diterapkan untuk menghilangkan sudut yang memiliki sudut yang bersebelahan dengan nilai V yang lebih tinggi. Beberapa definisi dari V adalah

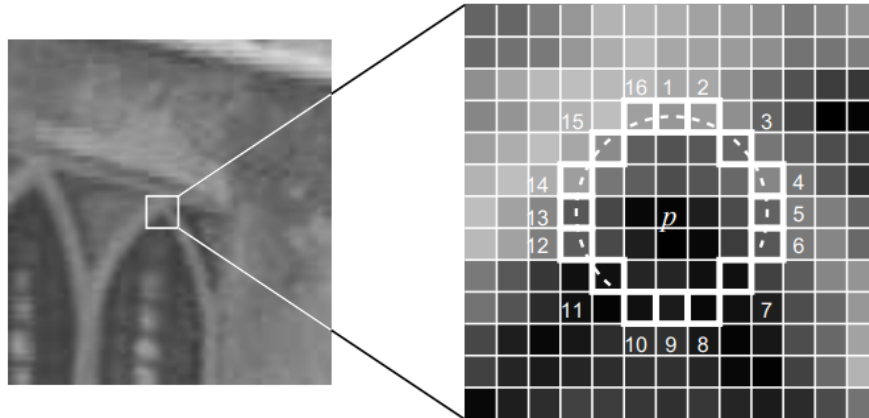
1. Maksimum nilai dari n untuk p yang masih merupakan sudut.
2. Maksimum nilai dari t untuk p yang masih merupakan sudut.
3. Jumlah dari perbedaan *absolute* antara piksel yang berdekatan di garis tepi dengan piksel yang berada di tengah.

Definisi 1 dan 2 sangat terstruktur dan banyak piksel yang berbagi nilai yang sama. Untuk perhitungan secara cepat dapat menggunakan definisi 3 dengan rumus: (Rosten & Drummond, 2006):

$$V = \max \left(\sum_{x \in S_{bright}} |I_{p \rightarrow x} - I_p| - t, \sum_{x \in S_{dark}} |I_p - I_{p \rightarrow x}| - t \right) \dots \dots \dots (2.6)$$

FAST merupakan algoritma deteksi sudut yang bekerja dengan memperhitungkan lingkaran 16 (enam belas) piksel di sekitar titik p. Langkah algoritma FAST adalah sebagai berikut : (Rosten & Drummond, 2006).

1. Menentukan titik p sebagai piksel yang akan diuji.
2. Menentukan *threshold* (t).
3. Tentukan 16 piksel yang berada disekitar titik p dimana radius lingkaran yang biasa digunakan adalah 3 yang berarti 3 piksel dari titik p, contoh pada Gambar 2.7.
4. Tentukan nilai n, nilai n akan digunakan dalam menentukan sudut dimana jumlah $n < 12$, nilai n untuk mendapatkan hasil terbaik adalah 9.
5. Pengujian dimulai dengan membandingkan piksel 1, 5, 9 dan 13 dari 16 piksel yang telah ditentukan sebelumnya dengan titik $p + t$ atau $p - t$.
6. Jika tiga piksel dari empat piksel pengujian tidak lebih besar dari $p + t$ atau tidak lebih kecil dari $p - t$ maka titik p bukan lanjutan dengan titik p berikutnya, jika iya maka lakukan pengurangan intensitas titik p dengan 16 piksel dan kemudian hasil pengurangan di jumlahkan.
7. Lakukan pengujian berulang untuk semua piksel pada citra.
8. Kemudian lakukan non-maximal suppression dengan mengecek apakah titik ke i dan titik i -1 lebih kecil sama dengan 4 atau tidak jika tidak lanjutkan ketitik selanjutnya. Jika iya maka bandingkan penjumlahan dari hasil pengurangan sebelumnya dari titik ke i dengan titik ke i-1. Jika jumlah i lebih besar dari i-1 maka titik ke i-1 bukan sudut jika iya maka titik ke - i bukan sudut.



Gambar 2.7 Deteksi sudut dengan algoritma FAST

(Sumber : Rosten & Drummond, 2006)

2.10 Speed Up Robust Features (SURF)

SURF merupakan algoritma untuk deteksi dan deskripsi *keypoint* dimana SURF adalah peningkatan dari algoritma *Scale-invariant feature transform* (SIFT) yang lebih cepat. SURF menggunakan penggabungan algoritma citra integral (*integral image*) dan *blob detection* yang berdasarkan determinan matriks *Hessian*. (Bay, et al., 2006)

Matriks *Hessian* adalah matriks yang setiap elemennya dibentuk dari turunan parsial kedua dari suatu fungsi). Misalkan $f(x)$ fungsi dengan n *variable* yang memiliki turunan parsial kedua dan turunannya kontinu, matrik *Hessian* $f(x)$ ditulis H adalah: (Leon, 1998)

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1^2 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_2} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix} \dots \dots \dots (2.19)$$

SURF dibagi menjadi beberapa tahapan sebagai berikut (Hironimus Gesu, et al, 2014):

1. *Interest Point Detection*

Deteksi *interest point* merupakan proses memilih titik-titik yang mengandung informasi dan yang stabil terhadap gangguan local maupun non-lokal dalam citra digital. Metode yang digunakan dalam pendeteksian adalah *blob detection* yaitu dengan menggunakan *Laplacian of Gaussian*. Rumus *gaussian* adalah sebagai berikut (Oyalon dan Rabin, 2015):

$$G(x, y) = \frac{1}{(2\pi\sigma^2)} e^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} \dots \dots \dots (2.20)$$

2. *Scale Space Representation*

Untuk perbedaan ukuran pada citra, algoritma SURF menggunakan metode perbandingan skala dimana metode ini menggunakan *scale space*, dimana citra di implementasikan dalam bentuk sebuah *image pyramid* (Lowe DG, 2004). Citra akan di melalui proses *smoothing* dengan fungsi *gaussian* dan secara berurutan dengan cara *sub-sampling* untuk mencapai tingkat tertinggi pada piramida. Dengan menggunakan *integral image*, perhitungan dapat menggunakan beberapa skala citra yang berbeda.

3. *Feature Description*

Feature merupakan bagian dari informasi yang dibutuhkan untuk pemrosesan. Tujuan dari tahap ini adalah untuk mendapatkan deskripsi dari fitur-fitur dalam citra yang diamati. Dalam mendeskripsikan fitur, SURF menggunakan *wavelet* dalam arah *vertikal* dan *horizontal*. Diambil tetangga terdekat dengan ukuran 20x20 di sekitar *keypoint*. Kemudian dibagi menjadi 4 bagian, setiap bagian diambil *vertikal* dan *horizontal wavelet* dan di jadikan sebuah *vector* dengan rumus dibawahini:

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|) \dots \dots \dots (2.21)$$

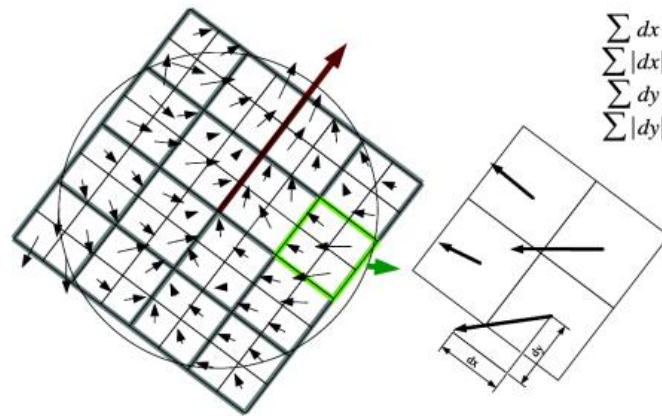
Dimana:

d_x : responhaar wavelet untuk arah *horizontal*

d_y : responhaar wavelet untuk arah *vertical*

$|dx|$: responhaar wavelet untuk arah *horizontal* yang di *absolute*

$|dy|$: responhaar wavelet untuk arah *vertical* yang di *absolute*



Gambar 2.8 Membangun *descriptor* dengan metode SURF

(Sumber : Bay, H., et al, 2006)

Berikut adalah rumus-rumus yang digunakan dalam mendeskripsi fitur dalam SURF (Oyalon dan Rabin, 2015):

- Konvolusi (O), Konvolusi merupakan operasi matematika yang mendasar dalam memproses citra. Konvolusi merupakan cara mengalikan dua *array* dengan ukuran yang berbeda dan dimensi yang sama untuk menghasilkan *array* baru dengan dimensi yang sama (Fisher et al, 2003). Berikut rumus dari konvolusi dalam SURF (Oyalon dan Rabin, 2015):

$$O(x, y) = Img(x - a - c, y - b - d) + Img(x - a, y - b) - Img(x - a - c, y - b) - Img(x - a, y - b - d) \dots \dots \dots (2.22)$$

- Haar wavelet* dalam SURF yang digunakan adalah menggunakan rumus konvolusi sehingga menghasilkan rumus *haar wavelet* dibawah (Oyalon dan Rabin, 2015):

$$H(x) = O(Img, a, b, c, d, x, y) + O(Img, a, b, c, d, x, y) \dots \dots \dots (2.23)$$

$$Hx(x, y) = -(O(Img, 1, -\lambda - 1, -\lambda - 1, 2\lambda + 1, x, y) + O(Img, 0, -\lambda - 1, \lambda + 1, 2\lambda + 1, x, y)) \dots \dots \dots (2.24)$$

$$Hx(x, y) = -(O(Img, -\lambda - 1, 1, 2\lambda + 1, -\lambda - 1, x, y) + O(Img, -\lambda - 1, 0, 2\lambda + 1, \lambda + 1, x, y)) \dots \dots \dots (2.25)$$

Tahapan dalam mendeskripsikan fitur adalah sebagai berikut (Oyalan dan Rabin, 2015)

- 1 $Cp = \cos$ (orientasi), $Sp = \sin$ (orientasi).
- 2 Lakukan perulangan dari $i = 0$ sampai $i = 3$.
- 3 Lakukan perulangan dari $j = 0$ sampai $j = 3$.
- 4 $m = 4*i + j$, $Dx[m] = 0$, $AbsDx[m]$, $Dy[m] = 0$, $AbsDy[m] = 0$.
- 5 Lakukan perulangan dari $k = 0$ sampai $k = 4$.
- 6 Lakukan perulangan dari $l = 0$ sampai $l = 4$.
- 7 $u = \text{keypoint} \rightarrow x + \text{skala} * (Cp*((i-2)*5 + k + 0,5)) - (Sp*((j-2)*5 + l + 0,5))$.
- 8 $v = \text{keypoint} \rightarrow y + \text{skala} * (Sp*((i-2)*5 + k + 0,5)) + (Cp*((j-2)*5 + l + 0,5))$.
- 9 $responx = \text{haarx}(Img, u, v, \text{skala}*0.5)$.
- 10 $respony = \text{haary}(Img, u, v, \text{skala}*0.5)$.
- 11 Lakukan perhitungan *gaussian* dengan rumus 2.24 dimana $x = (i-2)*5 + k + 0.5$, $y = (j-2)*5 + l + 0.5$, $\sigma = 3.3$.
- 12 $responu = \text{gauss} * (responx*Cp + respony*Sp)$.
- 13 $responv = \text{gauss} * (-responx*Sp + respony*Cp)$.
- 14 $Dx[m] += responu$, $AbsDx[m] += \text{abs}(responu)$, $Dy[m] += responv$, $AbsDy[m] += \text{abs}(responv)$.
- 15 Ulangi langkah 6 sampai 14.
- 16 Ulangi langkah 5 sampai 15.
- 17 $norm += \text{AbsDx}[m]^2 + \text{AbsDy}[m]^2 + Dx[m]^2 + Dy[m]^2$.
- 18 Ulangi langkah 3 sampai 17.
- 19 Ulangi langkah 2 sampai 18.
- 20 $norm = \text{sqrt}(norm)$.
- 21 Jika $norm$ tidak sama dengan 0 lakukan langkah 22 sampai 24
- 22 Lakukan perulangan $i = 0$ sampai $i = 15$.

23 $Dx[i] \neq norm, AbsDx \neq norm, Dy \neq norm, AbsDy \neq norm.$

24 Ulangi langkah 22 sampai 23.

4. *Feature Matching and Recognition*

Tahap ini membandingkan fitur hasil perhitungan pada proses sebelumnya tetapi hanya bila terdapat perbedaan kontras, yang di deteksi melalui tanda dari *trace* matriks *hessian*. Objek di wakili oleh vektor fitur, kesamaan dan perbedaan antara dua citra dapat diketahui dengan cara membandingkan antara urutan kedua vektor fitur tersebut untuk dapat melakukan klasifikasi citra. Secara umum metode untuk membandingkan citra dapat dilakukan melalui perbandingan jarak ke dua vektor fitur atau melalui kesamaan ke dua vektor fitur. Misalnya, dua citra di bandingkan dengan menghitung jarak antara dua vektor fitur, semakin pendek jarak maka lebih besar kesamaan dan lebih kecil perbedaan. Dimana untuk setiap *sub-region*, penjumlahan dari dx , $|dx|$, dy dan $|dy|$ dihitung berdasarkan pada orientasi *sample grid*, dimana dx dan $|dx|$ adalah x respon *wavelet Haar* pada arah *horizontal* dan dy dan $|dy|$ adala y respon *wavelet Haar* pada arah *vertikal* dan ke dua arah ini merujuk pada orientasi titik utama (*keypoint*).

2.11 Ekstraksi Fitur

Ekstraksi fitur merupakan bagian fundametal dari analisis citra. Fitur adalah karakteristis unik dari suatu objek. Karakteristik fitur yang baik memenuhi persyaratan dibawahini (Putra, 2010):

1. Dapat membedakan suatu objek dengan yang lainnya (*discrimination*).
2. Memperhatikan kompleksitas komputasi dalam memperoleh fitur. Kompleksitas komputasi yang tinggi akan menjadi sebuah kendala dalam menemukan suatu fitur.
3. Tidak terikat (*independence*) yang berarti bersifat *invariant* terhadap berbagai transformasi (rotasi, *scaling*, translasi, dan lain sebagainya).
4. Jumlahnya sedikit, karena fitur yang jumlahnya sedikit dapat menghemat waktu komputasi dan *storage* untuk proses selanjutnya.

2.12 Sistem Koordinat Warna

Warna atau sistem ruang warna yang diciptakan untuk memenuhi tujuan tertentu atau yang di ciptakan untuk platform perangkat keras tertentu. Menurut teori T .Young, setiap warna tertentu dapat diciptakan dengan menggabungkan (trsimulus) tiga warna utama C1,C2, dan C3 dengan persentasi tertentu. (Darma Putra, 2010)

$$C = aC1 + bC2 + cC3 \dots\dots\dots(2.1)$$

Teori ini didasarkan pada kenyataan bahwa mata manusia memiliki tiga bagian pada retina yang dapat menangkap puncak-puncak spektrum dari warna kuning-hijau, hijau, dan biru. Untuk nilai krominan dari sebuah warna adalah rasio dari sebuah warna tersebut yaitu .(Darma Putra, 2010)

$$c_i = \frac{c_i}{c_1 + c_2 + c_3}, i = 1,2,3 \dots\dots\dots(2.2)$$

2.13 Precision Dan Recall

Dalam melakukan pengenalan objek tidak semua hasil yang di tampilkan selalu benar, maka diperlukan tahapan untuk menguji hasil prediksi yang dilakukan. Metode yang dapat digunakan dalam pengujian adalah *recall* dan *precision*. *Recall* merupakan banyaknya hasil yang benar dari prediksi yang dilakukan. Sedangkan *precision* merupakan hasil prediksi yang dilakukan (Powers, 2007).

Precision dan *recall* akan menggunakan matriks hasil dari pencocokan dengan *threshold* seperti pada rumus sebagai berikut (Oyallon dan Rabin, 2015):

$$recall(t) = \frac{\#correct\ NN\ matches(t)}{\#correct\ matches(1)} \in [0,1] \quad (2.26)$$

$$precision(t) = \frac{\#correct\ NN\ matches(t)}{\#NN\ matches(t)} \in [0,1] \quad (2.27)$$

2.14 Penelitian Terdahulu

- **Penelitian Pertama – Faizal Zuli (2018)**

Pada penelitian ini dengan judul jurnal “*Rancangan Bangun Augmented dan Virtual Reality Menggunakan Algoritma FAST sebagai Media Informasi 3D di Universitas Satya Negara Indonesia*”. Pada penelitian ini dimana penerapan yang digunakan yaitu algoritma FAST dan juga algoritma SURF yang dimana pada Algoritma FAST digunakan untuk mengambil beberapa sudut dari suatu objek dan menyimpulkan isi dari suatu images dan algoritma SURF digunakan untuk menganali gambar.

- **Penelitian Kedua – Firma Firmansyah Adi, dkk (2017)**

Penelitian kedua ini dengan judul jurnal “*Implementasi Algoritma Speed Up Robust Features (SURF) Pada Pengenalan Rambu-rambu Lalu Lintas*”. Yang dimana penelitian ini menggunakan algoritma SURF bertujuan untuk ekstraksi fitur dengan cara mendeteksi fitur lokal suatu citra dengan handal dan cepat. Penelitian ini menghasilkan waktu pendeteksian rambu lalu lintas yang dapat di kenali dengan cukup cepat dengan waktu rata-rata pengenalan sebesar 101,2 milidetik. Selain itu Algoritma SURF dapat mendeteksi dalam berbagai ukuran dan sudut yang ditolerir dan bahkan memiliki kemampuan diskriminasi yang baik.

- **Penelitian Ketiga – Johar Nur Lin, dkk (2017)**

Selanjutnya penelitian ketiga dengan judul “*Sistem Deteksi dan Ekstraksi Keypoint dari Bagian Jenis Sayap Nyamuk Dengan Menggunakan Algoritma Speed Up Robust Features*”. Penelitian kali ini menunjukan rancangan bangun aplikasi deteksi dan ekstraksi keypoint dari gambar sayap nyamuk menggunakan algoritma *From Accelerated Segment Test* (FAST). Pada algoritma *Speed Up Robust Features* (SURF) menghasilkan tahap normalisasi yang disimpan, kemudian digunakan pada proses deteksi fitur lokal yang menarik (*keypoint*) dari gambar sayap nyamuk dideteksi. Normalisasi di gunakan untuk mengubah format *image* sayap nyamuk yang sebelumnya (JPG) menjadi (PNG) dengan ukuran 500x281 piksel.

- **Penelitian Keempat – Muhammad Baresi Ariel, dkk (2016)**

Berikutnya penelitian keempat ini dengan judul “*Implementasi Metode Speed Up Robust Features dan Scale Invariant Features Transform untuk Identifikasi Telapak Kaki Individu*”. Penelitian kali ini menjelaskan bahwa pada algoritma SURF mampu mendeteksi fitur lokal suatu citra dengan handal dan cepat. Algoritma ini terinspirasi dari *Scale Invariant Features Transform* (SIFT) terutama pada tahap *scale space representation*. Algoritma SURF menggunakan penggabungan algoritma integral *image* dan *blob detection* berdasarkan determinan dari matriks Hessian

2.15 Visual Studio, C#, XAMPP dan Mysql

Visual Studio 2010 pada dasarnya adalah sebuah bahasa pemrograman komputer. Dimana pengertian dari bahasa pemrograman itu adalah perintah-perintah atau instruksi yang dimengerti oleh komputer untuk melakukan tugas-tugas tertentu. Visual Studio 2010 (yang sering juga disebut dengan VB .Net 2010) selain disebut dengan bahasa pemrograman, juga sering disebut sebagai sarana (tool) untuk menghasilkan program-program aplikasi berbasis windows. (Ninuk,2017)

C# (dibaca: C sharp) merupakan sebuah bahasa pemrograman yang berorientasi objek yang dikembangkan oleh Microsoft sebagai bagian dari inisiatif kerangka .NET *Framework*. Bahasa pemrograman ini dibuat berbasis bahasa C++ yang telah dipengaruhi oleh aspek - aspek atau pun fitur bahasa yang terdapat pada bahasa - bahasa pemrograman lainnya seperti *Java*, *Delphi*, *Visual Basic* dan lain - lain dengan beberapa penyederhanaan. Menurut standar ECMA-334 C# *Language Specification*, nama C# terdiri atas sebuah huruf latin C (U+0043) yang diikuti oleh tanda pagar yang menandakan angka # (U+0023). Tanda pagar # yang digunakan memang bukan tanda kres dalam seni musik (U+266F), dan tanda pagar # (U+0023) tersebut digunakan karena karakter kres dalam seni musik tidak terdapat didalam keyboard standar(Adelia,dkk,2011)

XAMPP adalah *software web server apache* yang di dalamnya tertanam *server MySQL* yang didukung dengan bahasa pemrograman PHP untuk membuat *website* yang dinamis. XAMPP sendiri mendukung dua *system* operasi yaitu

windows dan Linux. Untuk linux dalam proses penginstalannya menggunakan *command line* sedangkan untuk windows dalam proses penginstalannya menggunakan *interface grafis* sehingga lebih mudah dalam penggunaan XAMPP di Windows di banding dengan Linux. XAMPP yang merupakan aplikasi vital bagi berjalannya XAMPP dengan baik. Di dalam XAMPP ada 3 komponen utama yang di tanam di dalamnya yaitu *web server Apache*, PHP, dan MySQL(Ninuk,2017)

MySQL dapat digunakan untuk membuat dan mengola database beserta isinya. Kita dapat memanfaatkan MySQL untuk menambahkan, mengubah dan menghapus data yang berada dalam database. MySQL merupakan sisitem manajemen *database* yang bersifat *at relational*. Artinya data-data yang dikelola dalam database akan diletakkan pada beberapa tabel yang terpisah sehingga manipulasi data akan menjadi jauh lebih cepat. MySQL dapat digunakan untuk mengelola database mulai dari yang kecil sampai dengan yang sangat besar. MySQL juga dapat menjalankan perintah-perintah *Structured Query Language* (SQL) untuk mengelola database-database yang ada di dalamnya. Hingga kini MySQL sudah berkembang hingga versi 5. MySQL 5 sudah mendukung *trigger* untuk memudahkan pengelolaan tabel dalam database.

BAB III

METODOLOGI PENELITIAN

3.1 Tempat Dan Waktu Penelitian

3.1.1 Tempat Penelitian

Pada penelitian kali ini untuk tempat penelitiannya berada di Lab Fakultas sains dan Teknologi UIN Sumatra Utara Medan

3.1.2 Waktu dan Jadwal Pelaksanaan Penelitian

Waktu pelaksanaan penelitian terhitung pada bulan maret – bulan juni 2020

No	Uraian	Maret				April				Mei				Juni			
		Mingguan															
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1	Persiapan Penelitian																
2	Perancangan																
3	Pelaksaaan Siklus																
4	Pelaksaaan Siklus 2																
5	Pelaksaaan Siklus 3																
6	Pengolahan data																
7	Penyusunan Laporan																

Table 3.1 Waktu Pelaksanaan Penelitian

3.2 Bahan dan Alat Penelitian

3.2.1 Perangkat Keras

Perangkat keras yang di gunakan untuk penelitian kali ini yaitu Laptop/Komputer sebagai bahan untuk penelitian dan menyusun laporan

3.2.2 Perangkat Lunak

Untuk Perangkat Lunak yang digunakan dalam penelitian kali ini yaitu C# (*Visual Studio*) untuk menjalankan program yang akan di lakukan dalam penelitian kali ini

3.3 Proses Kerja

Proses kerja dari penelitian pada perangkat lunak untuk pengenalan tempat dengan metode (algoritma) SURF dan FAST yang terdiri dari beberapa proses yaitu proses pencarian untuk tempat yang dimana *user* meng - *input* citra (gambar) yang akan dicocokkan dengan citra (gambar) yang terdapat pada *database* dan diambil citra (gambar) yang memiliki tingkat kecocokan yang paling tinggi, selanjutnya ada proses tampil data yang dimana pada proses kali ini menampilkan semua data tempat yang terdapat pada *database*, selanjutnya untuk yang terakhir adalah dimana proses kali ini melihat lokasi dimana *user* dapat melihat suatu lokasi dari hasil pencarian tersebut atau dari data citra yang terdapat pada *database*. Berikut penjelasannya dari setiap proses yang ada :

3.3.1 Proses Pencarian Tempat

Sebelum proses pencarian dilakukan, pada *database* terdapat 100 (seratus) data untuk suatu proses pencarian tempat yang diawali dengan *user* meng-*input* citra (gambar) dimana citra (gambar) seperti citra gedung, jalan, atau juga dapat seperti lapangan. Selanjutnya citra di *input* dan dikonversi ke citra *grayscale*. Untuk tahap selanjutnya perbaikan kontras pada citra dengan metode ekualisasi histogram. berikutnya lakukan pendeteksian sudut (fitur) dengan menggunakan metode FAST, lalu untuk berikutnya jika sudut sudah di deteksi dilanjutkan untuk melakukan deskripsi fitur dengan menggunakan

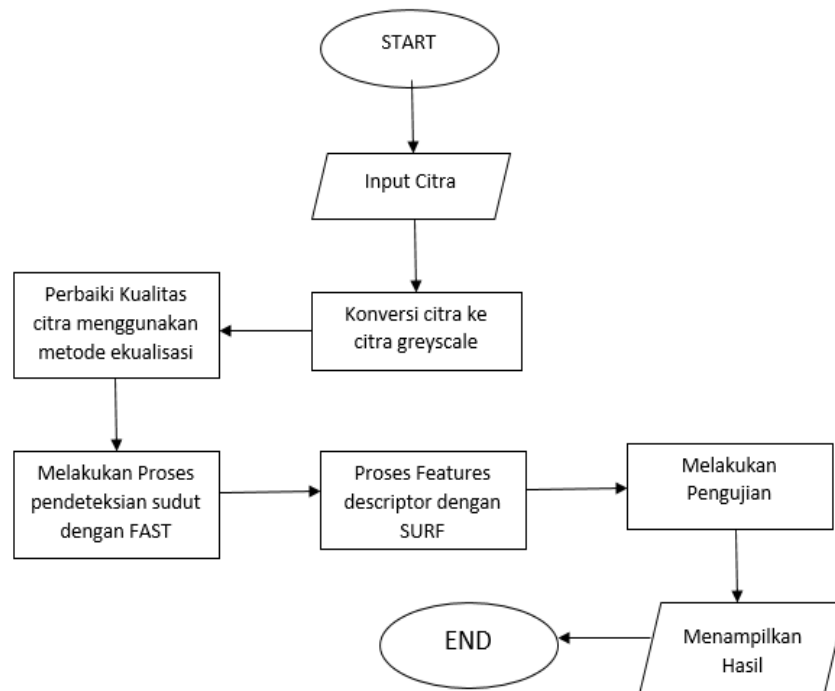
metode SURF yang menghasilkan *descriptor* dan orientasi dari setiap sudut. Lakukan hal yang sama untuk data citra yang terdapat pada *database*, dan tahap selanjutnya adalah lakukan perhitungan tingkat *error* dengan *precision* dan *recall*. Tahapan terakhir adalah menampilkan hasil pencarian.

1. Proses input citra

Pada Proses *input* citra kali ini pertama adalah akan terdapat dimana *user* akan melakukan *penginput* pada citra yang dimana citra dapat berupa seperti gedung, jalanan atau lapangan dan sebelumnya *user* akan melakukan *penginput* dan pada *database* sudah terdapat data seperti contoh berikut:



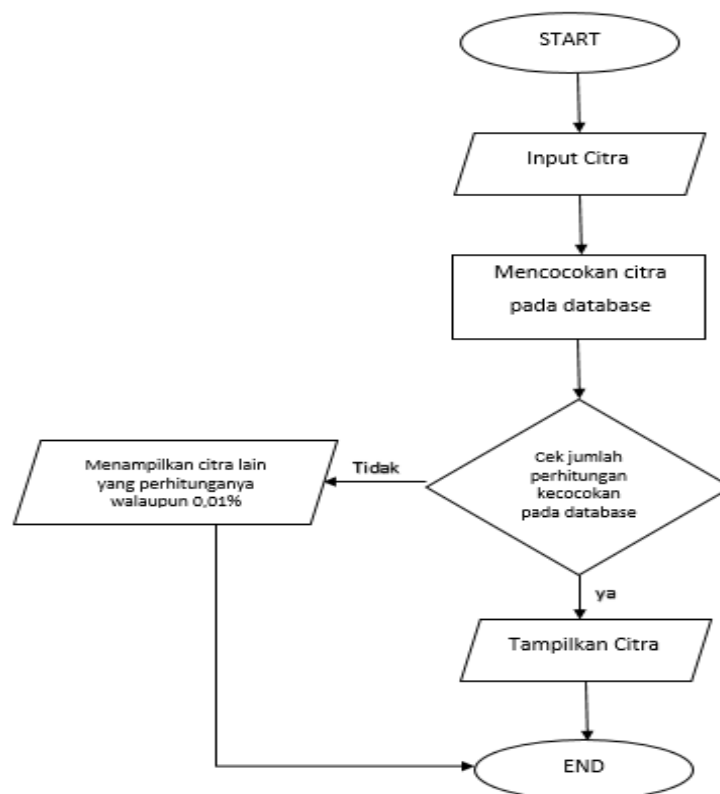
Gambar 3.1 Contoh Citra pada proses pencocokan



Gambar 3.2 Flowchart Pencarian Tempat

3.3.2 Proses Tampilan Data

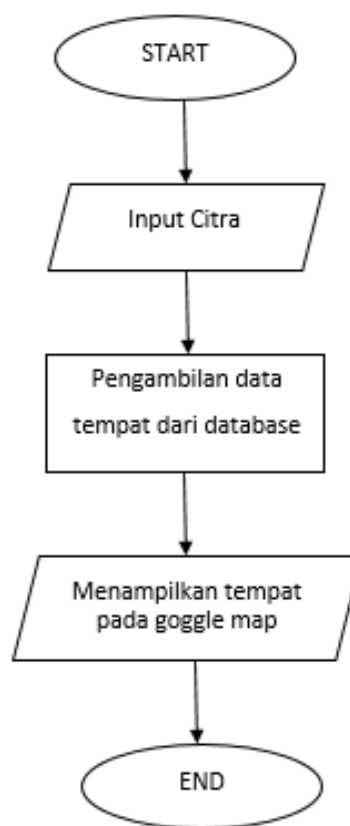
Untuk proses dalam menampilkan data dapat dilakukan saat *user* mengakses fitur untuk menampilkan data dimana pada sistem yang akan menampilkan seluruh data yang terdapat pada *database*. Selanjutnya untuk proses menampilkan data dapat dilakukan dengan mengambil semua data tempat yang terdapat pada *database* dan ditampung ke dalam *array*. Berikutnya lakukan perhitungan pada *array* dan lihat apakah data tersebut ada atau tidak. Jika pada *array* kosong maka akan menampilkan pesan kesalahan bahwa tidak ada data tempat pada *database*. Jika terdapat data maka akan dicetak data ke-*n*, setelah di cek apakah *n* lebih besar dari jumlah data di *array*, jika tidak maka *n* akan ditambah dengan 1 dan lakukan pencetakan kembali tahap ini akan berulang sampai *n* lebih besar dari jumlah data di *array*. Jika *n* sudah lebih besar dari jumlah data di *array* maka proses selesai. Berikut adalah *flowchart* dari proses tampil data:



Gambar 3.3 *Flowchart* Tampil Data

3.3.3 Proses Melihat Lokasi

Dalam melakukan proses kali ini yaitu untuk dapat melakukan penglihatan lokasi dapat dilakukan dengan cara pengambilan data *latitude* dan *longitude* tempat dari *database*. Selanjutnya akan ditampilkan melalui peta yang berisi lokasi - lokasi data dengan menggunakan google map yang ditanganin oleh oleh perusahaan ternama yaitu *google*.



Gambar 3.4 *Flowchart* Melihat Tempat

3.3.4 Proses Pendeteksian sudut dengan Metode FAST

Pada proses kali ini untuk mendapatkan pencarian tempat dengan menggunakan metode kali ini yaitu pada Tahap pendeteksian sudut menggunakan metode FAST. Metode FAST diawali dengan menginisialisasi tetap r , n dan t dimana r merupakan jari-jari terhadap pusat (p) yang berarti nilai

p dimulai dari titik $[r, r]$, n merupakan jumlah nilai yang akan dibandingkan dengan nilai p, nilai maksimal n adalah 12 dan t adalah *threshold* yang akan digunakan dalam perhitungan. Kemudian tahapan berikutnya yaitu melakukan perbandingan titik p dengan nilai pada titik 1, 3, 9 dan 13. Untuk tahap berikutnya mempunyai syarat selanjutnya yaitu tiga dari empat titik tersebut harus lebih besar dari $p + t$ atau lebih kecil dari $p - t$. Jika syarat tersebut tidak terpenuhi maka akan dilanjutkan ke titik p yang selanjutnya. Jika sudah memenuhi persyaratan sebelumnya maka titik p akan di bandingkan dengan titik, 16 titik di sekitar titik p jika jumlah pada nilai yang memenuhi persyaratan yang sama pada tahapan sebelumnya sama atau lebih besar dari n maka titik p dianggap sudut, jika tidak memenuhi syarat maka titik p bukan sudut.

3.3.5 Proses Fitur dengan metode SURF

Proses kali ini yaitu mendeskripsikan fitur dengan metode SURF. Dalam menentukan *descriptor*, pertama lakukan perhitungan orientasi, dalam menentukan orientasi maka lakukan inisialisasi skala, sektor dan L dimana nilai L adalah 7 kemudian lakukan tahapan algoritma dengan menggunakan rumus :

$$h(v) = \text{round} \left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (BatasAtas - BatasBawah) \right) +$$

BatasBawah. Perhitungan dilakukan pada setiap titik yang sudah di dapatkan pada tahap sebelumnya yaitu pada tahapan FAST dengan mengambil 20x20 tetangga terdekat dan dibagi menjadi 4 bagian.

3.3.6 Analisis Kebutuhan

Analisis kebutuhan merupakan tahapan yang penting dalam rekayasa perangkat lunak yang akan memperoleh hasil analisis yang baik serta dalam suatu sistem yang akan diteliti harus perlu dipahami hingga terperinci agar dapat terciptanya suatu *efisiensi* dan *efektivitas* dalam pembuatan maupun dalam berjalannya suatu *system* tersebut. Adapun untuk analisi kebutuhan dalam kinerja informasi, ekonomi serta pelayanan disini yaitu :

- 1 Aplikasi dapat menelusuri database untuk mencari letak citra pada citra yang akan di input.
- 2 Aplikasi dapat membantu mendeteksi lokasi yang akan ingin di cari sesuai dengan citra yang ada di dalam *database*.
- 3 Aplikasi tidak perlu harus menambahkan *software* lainnya agar dapat di jalankan.
- 4 Terdapat pada aplikasi yang menyediakan fasilitas untuk menginput lokasi yang belum terdapat pada *database*
- 5 Pada sistem aplikasi menyediakan citra di dalam *database* sehingga untuk *user* dapat melakukan mengujian terhadap *algoritma* dengan memberikan *input* berupa citra untuk di cocokan .
- 6 Pada sisitem *control* aplikasi disertai dengan validasi yang akan bekerja jika terjadinya kesalahan akan menampilkan kotak pesan informasi yang *valid*.

3.3.7 Perancangan

Program Pengenalan Tempat					
Cari	Tampilan	Pengujian	Tentang	Keluar	
<p style="text-align: center;"> Penerapan Metode SPEEDED UP ROBUST FEATURES (SURF) dan FEATURES FROM ACCELERATED SEGMENT TEST (FAST) untuk Pengelana Tempat </p> <p style="text-align: center;"> Mey Hendra Putra Sirait 0701163068 Sains dan Teknologi Ilmu Komputer Universitas Islam Negeri Sumatera Utara Medan </p>					

Gambar 3.5 Tampilan Pada *Form* Pertama

Program Pengenalan Tempat				
Cari	Tampilan	Pengujian	Tentang	Keluar



Gambar 3.6 *Form Pertama Pada Menu*

1. Pada menu terdapat didalamnya ada 'Proses', 'Tampilan' dan serta 'Pengujian' untuk menu pada 'Proses' adalah pemrosesan gambar yang sudah di *inputkan* serta akan di prosesnya gambar yang telah di *input*. Untuk menu 'Tampilan' adalah menampilkan gambar yang akan tesimpan di *database* dan untuk *menu* 'Pengujian' untuk menampilkan pengujian pada *form* tersebut.
2. Pada *About* adalah data (biodata) pembuat program tersebut.
3. Dan yang terkahir *Exit* adalah untuk 'keluar'.

Program Pengenalan Tempat				
Cari	Tampilan	Pengujian	Tentang	Keluar
<div style="display: flex; justify-content: space-around; margin-bottom: 20px;"> <div style="border: 1px solid black; padding: 5px 15px; text-align: center;">Masukan Gambar</div> <div style="border: 1px solid black; padding: 5px 15px; text-align: center;">Proses</div> </div> <div style="text-align: center; margin-bottom: 10px;">HASIL</div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; width: 150px; height: 80px; margin: 0 auto;"></div> <div style="border: 1px solid black; width: 150px; height: 80px; margin: 0 auto;"></div> </div> <div style="text-align: center; margin-bottom: 20px;">Keterangan :</div> <div style="text-align: center;"> <div style="border: 1px solid black; padding: 5px 15px; text-align: center;">Show Location</div> </div>				

Gambar 3.7 *Form Untuk Proses*

1. Pada *Button Input* gambar adalah gambar yang ingin diketahui lokasinya.
2. Pada *Button Proses* yaitu data akan di cocokan dengan citra di *database* untuk pencocokan.
3. Hasil akan di tampilkan lewat *picture box* yang di proses.
4. Keterangan akan memberikan informasi kecocokan dari proses.
5. *Picture box* pertama adalah gambar yang di inputkan dan untuk *picture box* ke 2 yaitu hasil yang di proses.
6. *Show location* adalah menampilkan lokasi .

Program Pengenalan Tempat				
Cari	Tampilan	Pengujian	Tentang	Keluar
<div><div><p>Keterangan :</p><p>Show Location :</p></div><div><p>Keterangan :</p><p>Show Location :</p></div></div>				

Gambar 3.8 *Form* Pada Tampilan

Pada *form* kali ini adalah untuk menampilkan citra *database* yang telah di proses atau sudah di *save* pada *database* tersebut dan menginformasikan keterangan citra yang telah di proses serta menampilkan hasil lokasi yang diberikan.

Program Pengenalan Tempat				
Cari	Tampilan	Pengujian	Tentang	Keluar
<p>Proses Pengujian :</p> <div style="border: 1px solid black; width: 150px; height: 60px; margin: 10px auto; text-align: center; line-height: 60px;">EXCEL</div>				
No	Citra Awal	Citra Hasil	Precision	Recall

Gambar 3.9 *Form Pada Pengujian*

Pada *form* pengujian kali ini untuk hasil proses (proses pengujian). Untuk yang tertera pada *microsoft excel* adalah data untuk *export* yang diterapkan pada aplikasi *microsoft excel* dan untuk selanjutnya pada tabel yang tertera yaitu menampilkan data citra awal ke citra hasil dari pengujian *presicion* dan *recall*.

BAB IV

HASIL DAN PEMBAHASAN

4.1 Pembahasan

4.1.1 Analisis Data

Pada bab ini akan di jelaskan mengenai hasil dan pengujian dari aplikasi pengenalan tempat yang menggunakan metode SURF dan FAST. Aplikasi yang menggunakan bahasa *c#* dan menggunakan basis *database* yang tersimpan d *mysql*. Data yang digunakan adalah data yang diambil dari hasil penelitian pada gedung sekitar area kampus 1 Universitas Islam Negeri Sumatra Utara. Pengujian dilakukan dengan menggunakan metode SURF dan FAST serta menggunakan *precision* dan *recall*, dimana pengujian dilakukan beberapa kali dengan beberapa citra input yang berbeda dari segi warna, orientasi, ukuran dan resolusi.

4.1.2 Representasi Data

Tahapan kali ini, untuk perhitungan pada dimana *user* melakukan input citra berupa gedung yang tertera untuk melakukan penginputan yang dimana *database* sudah terdapat data sebagai contoh pada gambar 4.1



Gambar 4.1 Potongan Citra yang di ambil

Pada kali ini proses yang akan di lakukan adalah mengubah citra RGB mejadi citra *grayscale* berikut perhitungan citra RGB menjadi citra *grayscale* dengan mengambil potongan citra input di atas dengan ukuran 8 x 8

101,153,213	52,101,157	52,101,157	52,101,157	52,101,157	52,101,157	23,67,116	23,67,116
77,125,174	20,64,111	20,64,111	20,64,111	20,64,111	20,64,111	15,53,92	15,53,92
77,125,174	20,64,111	20,64,111	20,64,111	20,64,111	20,64,111	15,53,92	15,53,92
77,125,174	20,64,111	20,64,111	20,64,111	20,64,111	20,64,111	15,53,92	15,53,92
77,125,174	20,64,111	20,64,111	20,64,111	20,64,111	20,64,111	15,53,92	15,53,92
80,125,167	47,86,125	47,86,125	47,86,125	47,86,125	47,86,125	58,92,120	58,92,120
80,125,167	47,86,125	47,86,125	47,86,125	47,86,125	47,86,125	58,92,120	58,92,120
80,125,167	47,86,125	47,86,125	47,86,125	47,86,125	47,86,125	58,92,120	58,92,120

Table 4.1 nilai RGB dari potongan citra

Untuk setiap piksel dari matriks di atas adalah sebagai berikut :

$$X[] = 0,299 * \text{citra grayscale} + 0,587 * \text{citra grayscale} + 0,114 * \text{citra grayscale} \\ = ?$$

$$x[0,0] = 0,299 * 101 + 0,587 * 153 + 0,114 * 213 = 144$$

$$x[0,1] = 0,299 * 52 + 0,587 * 101 + 0,114 * 157 = 93$$

$$x[0,2] = 0,299 * 52 + 0,587 * 101 + 0,114 * 157 = 93$$

$$x[0,3] = 0,299 * 52 + 0,587 * 101 + 0,114 * 157 = 93$$

$$x[0,4] = 0,299 * 52 + 0,587 * 101 + 0,114 * 157 = 93$$

$$x[0,5] = 0,299 * 52 + 0,587 * 101 + 0,114 * 157 = 93$$

$$x[0,6] = 0,299 * 23 + 0,587 * 67 + 0,114 * 116 = 59$$

$$x[0,7] = 0,299 * 23 + 0,587 * 67 + 0,114 * 116 = 59$$

$$x[1,0] = 0,299 * 77 + 0,587 * 125 + 0,114 * 174 = 116$$

$$x[1,1] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[1,2] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[1,3] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[1,4] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[1,5] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[1,6] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[1,7] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[2,0] = 0,299 * 77 + 0,587 * 125 + 0,114 * 174 = 116$$

$$x[2,1] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[2,2] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[2,3] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[2,4] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[2,5] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[2,6] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[2,7] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[3,0] = 0,299 * 77 + 0,587 * 125 + 0,114 * 174 = 116$$

$$x[3,1] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[3,2] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[3,3] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[3,4] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[3,5] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[3,6] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[3,7] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[4,0] = 0,299 * 77 + 0,587 * 125 + 0,114 * 174 = 116$$

$$x[4,1] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[4,2] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[4,3] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[4,4] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[4,5] = 0,299 * 20 + 0,587 * 64 + 0,114 * 111 = 56$$

$$x[4,6] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[4,7] = 0,299 * 15 + 0,587 * 53 + 0,114 * 92 = 46$$

$$x[5,0] = 0,299 * 80 + 0,587 * 125 + 0,114 * 167 = 116$$

$$x[5,1] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[5,2] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[5,3] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[5,4] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[5,5] = 0,299 * 47 + 0,587 * 64 + 0,114 * 125 = 79$$

$$x[5,6] = 0,299 * 58 + 0,587 * 92 + 0,114 * 102 = 85$$

$$x[5,7] = 0,299 * 58 + 0,587 * 92 + 0,114 * 102 = 85$$

$$x[6,0] = 0,299 * 80 + 0,587 * 125 + 0,114 * 167 = 116$$

$$x[6,1] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[6,2] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[6,3] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[6,4] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[6,5] = 0,299 * 47 + 0,587 * 64 + 0,114 * 125 = 79$$

$$x[6,6] = 0,299 * 58 + 0,587 * 92 + 0,114 * 102 = 85$$

$$x[6,7] = 0,299 * 58 + 0,587 * 92 + 0,114 * 102 = 85$$

$$x[7,0] = 0,299 * 80 + 0,587 * 125 + 0,114 * 167 = 116$$

$$x[7,1] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[7,2] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[7,3] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[7,4] = 0,299 * 47 + 0,587 * 86 + 0,114 * 125 = 79$$

$$x[7,5] = 0,299 * 47 + 0,587 * 64 + 0,114 * 125 = 79$$

$$x[7,6] = 0,299 * 58 + 0,587 * 92 + 0,114 * 102 = 85$$

$$x[7,7] = 0,299 * 58 + 0,587 * 92 + 0,114 * 102 = 85$$

Dari perhitungan diatas menghasilkan citra *grayscale* dengan matriks sebagai berikut:

144	93	93	93	93	93	59	59
116	56	56	56	56	56	46	46
116	56	56	56	56	56	46	46
116	56	56	56	56	56	46	46
116	56	56	56	56	56	46	46
116	79	79	79	79	79	85	85
116	79	79	79	79	79	85	85
116	79	79	79	79	79	85	85

Table 4.2 Nilai Citra setelah Konverksi *Grayscale*

Dan berikutnya lanjut pada perhitungan histogramnya sebagai berikut :

- a Batas akhir = nilai maksimal dari matriks, dan batas bawah = 0.
- b Lakukan pengulangan dengan $n = \text{batas bawah} \text{ hingga } \leq \text{batas akhir}$ dengan $\text{step } n + 1$.
- c Inisialisasi $k = n$.
- d Lakukan perhitungan ini jumlah piksel dengan nilai n .
- e Lakukan perhitungan nilai $rk = n / (\text{batas akhir})$.
- f Lakukan perhitungan nilai $P = ni / (\text{jumlah piksel citra})$.
- g Setelah perulangan selesai, lakukan perbandingan P dengan nilai rk yang terdekat dan dapatkan nilai $x = k$.
- h Lakukan perubahan nilai *pixel* k menjadi x .

K	Jlh Piksel	$r(k) = k /$ batasatas	$P = \text{jlhpiksel} /$ (baris * kolommatriks)
0	0	0	0
1	0	0,00694	0
2	0	0,01389	0
3	0	0,02083	0
4	0	0,02778	0
5	0	0,03472	0
...
138	0	0,95833	0
139	0	0,96528	0
140	0	0,97222	0
141	0	0,97917	0
142	0	0,98611	0
143	0	0,99306	0
144	1	1	0,015625

Table 4.3 Nilai Citra untuk Proses Histogram *Equalization*

Dari perhitungan yang tertera di atas maka di dapatkan hasil sebagai berikut :

$$k = 46, P = 0,125$$

$$k = 56, P = 0,3125$$

$$k = 59, P = 0,03125$$

$$k = 79, P = 0,234375$$

$$k = 86, P = 0,09375$$

$$k = 116, P = 0,109375$$

$$k = 144, P = 0,015625$$

Dari hasil P diatas maka ubah nilai k dari nilai P dengan nilai piksel k yang baru dimana nilai piksel k yang baru didapatkan melalui perbandingan nilai P dengan nilai $r(k)$ terdekat dan didapatkan hasil sebagai berikut:

2	11	11	11	11	11	5	5
16	45	45	45	45	45	18	18
16	45	45	45	45	45	18	18
16	45	45	45	45	45	18	18
16	45	45	45	45	45	18	18
16	34	34	34	34	34	13	13
16	34	34	34	34	34	13	13
16	34	34	34	34	34	13	13

Table 4.4 Nilai Citra setelah Proses Histogram *Equalization*

Selanjutnya proses pendeteksian sudut dengan metode FAST sebagai berikut :

2	11	11	11	11	11	5	5
16	45	45	45	45	45	18	18
16	45	45	45	45	45	18	18
16	45	45	45	45	45	18	18
16	45	45	45	45	45	18	18
16	34	34	34	34	34	13	13
16	34	34	34	34	34	13	13
16	34	34	34	34	34	13	13

Table 4.5 Deteksi sudut dengan metode FAST

- a. Lakukan perbandingan p dengan titik 1, 3, 9, dan 13:

$P + 50 > \text{table di atas dengan } p : 45$

$$45 + 50 > 11 = \text{true}$$

$$45 + 50 > 18 = \text{true}$$

$$45 + 50 > 34 = \text{true}$$

$$45 + 50 > 16 = \text{true}$$

Dari hasil diatas adalah bahwa nilai 4 titik tersebut lebih kecil dari $p + t$ dan memenuhi syarat minimum 3 titik maka tahap akan dilanjutkan ketahap selanjutnya.

- b. Tahap ini titik p akan di bandingkan dengan titik 1 sampai 16 dan jika sebanyak n dari 16 titik yang di bandingkan lebih besar dari p+t atau lebih kecil p-t maka titik p merupakan sudut jika tidak, maka titik p bukan sudut:

$$45 + 50 > 11 = \text{true}$$

$$45 + 50 > 11 = \text{true}$$

$$45 + 50 > 45 = \text{true}$$

$$45 + 50 > 18 = \text{true}$$

$$45 + 50 > 18 = \text{true}$$

$$45 + 50 > 18 = \text{true}$$

$$45 + 50 > 34 = \text{true}$$

$$45 + 50 > 34 = \text{true}$$

$$45 + 50 > 34 = \text{true}$$

$$45 + 50 > 34 = \text{true}$$

$$45 + 50 > 34 = \text{true}$$

$$45 + 50 > 16 = \text{true}$$

$$45 + 50 > 16 = \text{true}$$

$$45 + 50 > 16 = \text{true}$$

$$45 + 50 > 45 = \text{true}$$

$$45 + 50 > 11 = \text{true}$$

Jumlah *true* adalah 16, maka lakukan pengurangan titik p dengan 16 titik sekitarnya :

$$\begin{aligned} \text{sum} = & \text{Abs}(45-11) + \text{Abs}(45-11) + \text{Abs}(45-45) + \text{Abs}(45-18) + \text{Abs}(45-18) + \\ & \text{Abs}(45-18) + \text{Abs}(45-34) + \text{Abs}(45-34) + \text{Abs}(45-34) + \text{Abs}(45-34) + \text{Abs}(45-34) + \\ & \text{Abs}(45-16) + \text{Abs}(45-16) + \text{Abs}(45-16) + \text{Abs}(45-45) + \text{Abs}(45-11) \end{aligned}$$

Lakukan sampai perulangan selesai.

- c. Misalkan hasil titik yang di dihasilkan adalah x[3,3] dan x[3,4] dengan *score* 10 dan 15 maka lakukan perbandingan sudut : $(3-3)*2 + (3-4)*2 = -2 \leq 4$, Benar maka lakukan perbandingan *sum* titik pertama dan titik kedua : $10 > 15$, salah maka titik x[3,3] bukan sudut.

Berikutnya tahap terakhir yaitu pendeskripsian fitur dengan metode SURF

- a. Tahapan pertama adalah mencari orientasi namun pada tahapan FAST sebelumnya orientasi yang di dapatkan adalah -1 karena pada FAST tidak memiliki tahapan untuk mencari orientasi.
- b. Tahapan selanjutnya adalah mencari vektor untuk *descriptor*:

$$Cp = \cos(\text{orientasi}) = 0.54, Sp = \sin(\text{orientasi}) = -0.84$$

$$\text{skala} = 0.4 * L = 0.4 * 7 = 2.8$$

$$i = 0$$

$$j = 0$$

$$\text{sum } dx[4*i+j] = 0$$

$$\text{Abs sum } dx[4*i+j] = 0$$

$$\text{Sum } dy[4*i+j] = 0$$

$$\text{Abs sum } dy[4*i+j] = 0$$

$$k = 0$$

$$l = 0$$

$$u = 3 + 2.8 * (0.54 * ((0-2)*5+0+0.5) - (-0.84) * ((0-2)*5+0+0.5)) = 33.755$$

$$v = 3 + 2.8 * (-0.84 * ((0-2)*5+0+0.5) - 0.54 * ((0-2)*5+0+0.5)) = 11$$

$$\text{haarx} = 0$$

$$\text{haary} = 0$$

$$\text{gauss} = 0.000004$$

$$\text{respon } u = 0.000004 * (0 * 0.011 + 0 * 0.99) = 0$$

$$\text{respon } v = 0.000004 * (-0 * 0.99 + 0 * 0.011) = 0$$

$$\text{sum } dx[0] += \text{respon } u = 0$$

$$\text{Abs sum } dx[0] += \text{abs}(\text{respon } u) = 0$$

$$\text{Sum } dy[0] += \text{respon } v = 0$$

$$\text{Abs sum } dy[0] += \text{abs}(\text{respon } v) = 0$$

$$\text{norm} += \text{sum } dx[0]^2 + \text{abs sum } dx[0]^2 + \text{sum } dy[0]^2 + \text{abs sum } dy[0]^2$$

Lakukan perulangan sampai $l = 4$, $k = 4$, $j = 3$ dan $i = 3$ dan hasil dari perhitungan diatas adalah sebagai berikut:

No	SumDx	AbsSumDx	SumDy	AbsSumDy
1	0	0	0	0
2	0,351494	0,351494	-0,22569	0,225692
3	0,243343	0,243343	-0,15625	0,156249
4	0	0	0	0
5	-0,05164	0,051644	0,122933	0,122933
6	3,029465	3,029465	-0,77257	1,721339
7	-0,64671	0,646705	0,400774	0,875656
8	-0,05164	0,051644	0,122933	0,122933
9	-0,10291	0,102914	0,244148	0,244148
10	2,374623	3,143898	-2,7544	3,658733
11	1,250696	2,258224	0,641475	2,028462
12	-0,10291	0,102914	0,244148	0,244148
13	0	0	0	0
14	0,607611	0,607611	-0,39014	0,390143
15	0,26966	0,26966	-0,17315	0,173146
16	0	0	0	0

Table 4.6 Pencarian Nilai Vektor untuk *Descriptor* dalam Ekstraksi Fitur

$$norm = 73,48135$$

$$norm = \sqrt{73,48135} = 8.572$$

$$norm \neq 0 = true$$

$$i = 0$$

$$sum\ dx[0] = 0 / 8.572 = 0$$

$$abs\ sum\ dx[0] = 0 / 8.572 = 0$$

$$sum\ dy[0] = 0 / 8.572 = 0$$

$$abs\ sum\ dy[0] = 0 / 8.572 = 0$$

Lakukan perulangan sampai $i = 15$ dan hasil akhir dari perhitungan diatas adalah sebagai berikut :

<i>No</i>	<i>SumDx</i>	<i>Abs Sum Dx</i>	<i>Sum Dy</i>	<i>Abs Sum Dy</i>
1	0	0	0	0
2	0,041004	0,041004	-0,02633	0,026329
3	0,028388	0,028388	-0,01823	0,018228
4	0	0	0	0
5	-0,00602	0,006025	0,014341	0,014341
6	0,353409	0,353409	-0,09013	0,200807
7	-0,07544	0,075443	0,046753	0,102152
8	-0,00602	0,006025	0,014341	0,014341
9	-0,01201	0,012006	0,028482	0,028482
10	0,277017	0,366758	-0,32132	0,426817
11	0,145903	0,263438	0,074833	0,236635
12	-0,01201	0,012006	0,028482	0,028482
13	0	0	0	0
14	0,070882	0,070882	-0,04551	0,045513
15	0,031458	0,031458	-0,0202	0,020199
16	0	0	0	0

Table 4.7 Nilai Vektor untuk *descriptor*

Tahapan ekstraksi *descriptor* diatas dilakukan pada setiap titik fitur yang telah di dapatkan sebelumnya. Kemudian lakukan tahapan diatas dari awal untuk citra dari *database* yang akan di cocokkan.

No	SumDx	AbsSumDx	SumDy	AbsSumDy
1	0	0	0	0
2	0,041004	0,041004	-0,02633	0,026329
3	0,028388	0,028388	-0,01823	0,018228
4	0	0	0	0

5	-0,00602	0,006025	0,014341	0,014341
6	0,353409	0,353409	-0,09013	0,200807
7	-0,07544	0,075443	0,046753	0,102152
8	-0,00602	0,006025	0,014341	0,014341
9	-0,01201	0,012006	0,028482	0,028482
10	0,277017	0,366758	-0,32132	0,426817
11	0,145903	0,263438	0,074833	0,236635
12	-0,01201	0,012006	0,028482	0,028482
13	0	0	0	0
14	0,070882	0,070882	-0,04551	0,045513
15	0,031458	0,031458	-0,0202	0,020199
16	0	0	0	0

Tabel 4.8 Deskriptor citra *input* pada titik a pada sistem koordinat warna

No	SumDx	AbsSumDx	SumDy	AbsSumDy
1	0	0	0	0
2	0,041004	0,041004	-0,02633	0,026329
3	0,028388	0,028388	-0,01823	0,018228
4	0	0	0	0
5	-0,00602	0,006025	0,014341	0,014341
6	0,353409	0,353409	-0,09013	0,200807
7	-0,07544	0,075443	0,046753	0,102152
8	-0,00602	0,006025	0,014341	0,014341
9	-0,01201	0,012006	0,028482	0,028482
10	0,277017	0,366758	-0,32132	0,426817
11	0,145903	0,263438	0,074833	0,236635
12	-0,01201	0,012006	0,028482	0,028482
13	0	0	0	0
14	0,070882	0,070882	-0,04551	0,045513
15	0,031458	0,031458	-0,0202	0,020199

16	0	0	0	0
----	---	---	---	---

Tabel 4.9 Deskriptor citra *database* pada titik a untuk koordinat warna

Lakukan perhitungan euclidean distance untuk semua deskriptor dari kedua citra dan misalkan hasil dari perhitungan euclidean distance tersebut adalah sebagai berikut:

	Titik A (x)	Titik B (x)	Titik C (x)	Titik D (x)
Titik A (y)	3.0042	1,0410	2,0143	0
Titik B (y)	2,0410	2,0342	4.8123	7.4252
Titik C (y)	0	3,0283	1,0183	6,0023
Titik D (y)	0	4.4251	5.9032	10,0932
Titik E (y)	0,0062	6,6025	6,4104	13,0143

Tabel 4.10 Hasil *euclidean distance* pada sistem kordinat warna

Dari tabel diatas maka ambil 2 deskriptor terkecil dari setiap titik seperti titik A pada citra y lebih dekat dengan titik D dan titik B pada citra x. Maka selanjutnya lakukan perbandingan $0 / 1.0410 < 0.8 = true$ maka titik A cocok dengan titik D. Pengambilan titik berdasarkan deskriptor terkecil. Lakukan proses yang sama untuk semua titik maka hasil pencocokan yang didapatkan adalah sebagai berikut:

No	Citra x	Citra Y
1	D	A
2	-	B
3	A	C
4	A	D
5	A	E

Tabel 4.11 Hasil pencocokan

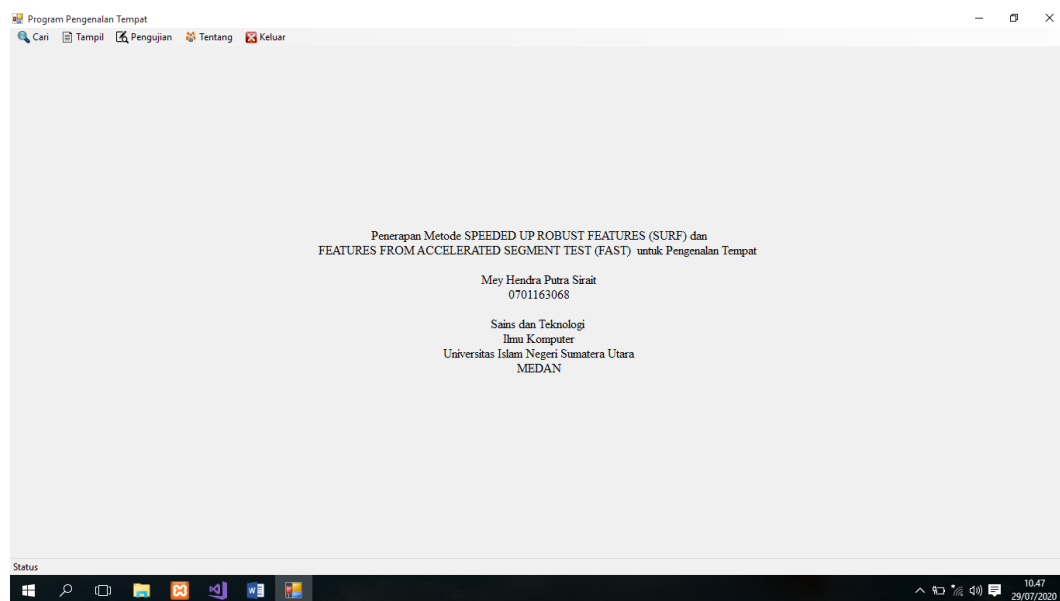
Dan terakhir pencocok pada *Precision* = hasil dari titik yang cocok / hasil dari seluruh titik fitur = $15 / 50 = 0.3$.

Recall = hasil dari titik yang cocok / jumlah titik yang cocok = $15 / 20 = 0.75$.

4.1.3 Hasil Analisis Data

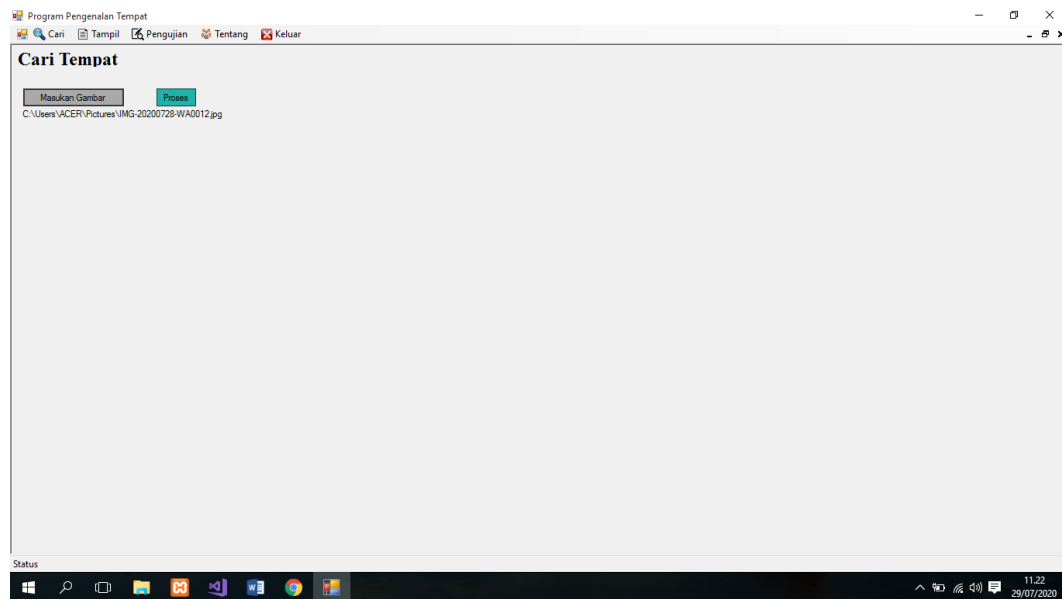
Hasil yang akan di tampilkan untuk mengetahui gedung pada citra yang telah di input akan mencari kococokan pada citra yang tertera pada *database* dengan menggunakan metode SURF dan FAST pada program yang telah dibuat.

4.1.4 Perancangan

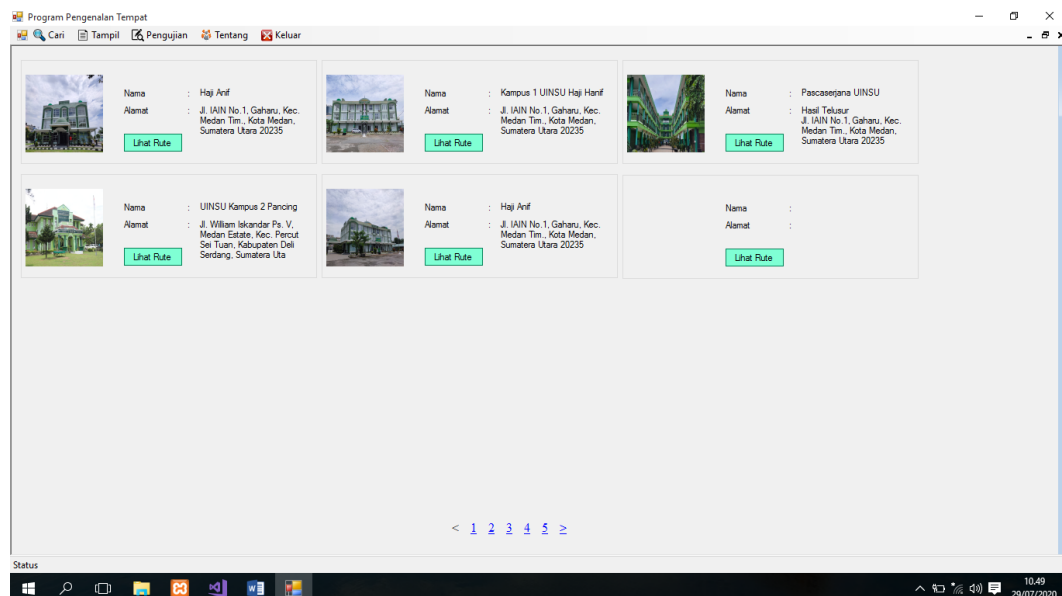


Gambar 4.2 *Form Awal*

Untuk *form* awal adalah halaman utama untuk menjalankan program yang akan di kerjakan. *Form* ini berisi pada halaman utamanya yang berupa judul skripsi, nama, serta universitas atau kata pembukaan dalam program ini.

Gambar 4.3 *Form Cari*

Pada *form* cari ini adalah proses untuk pengenalan tempat yang akan di *inputkan* yang akan dilakukan pencocokan terhadap citra yang akan di *inputkan*.

Gambar 4.4 *Menu Tampil*

Pada *form* ini (*form* tampil) adalah tampilan data yang tersimpan pada *database* yang digunakan untuk *inputan* citra yang akan di cocokan dan memiliki *rute* atau lokasi yang akan di tampilkan ketika ingin mengetahui lokasi tersebut.


No	Input	Hasil	Nama	Alamat	Precision	Recall	Waktu (s)
1	/src/img/Data/fh...	/src/img/Input/A...	Haji Anif	Jl. IAIN No. 1. Ga...	0.26464	0.26464	2047
2	/src/img/Data/P...	/src/img/Input/P...	Pascasejana UI...	Hasil Telusur.Jl. I...	0.0447844	0.0447844	49
3	/src/img/Data/fh...	/src/img/Input/K...	Kampus 1 UINS...	Jl. IAIN No. 1. Ga...	0.0371683	0.0371683	40
4	/src/img/Data/fh...	/src/img/Input/...	Haji Anif	Jl. IAIN No. 1. Ga...	0.0412569	0.0412569	40
5	/src/img/Data/fh...	/src/img/Input/...	Haji Anif	Jl. IAIN No. 1. Ga...	0.0412569	0.0412569	27
6	/src/img/Data/P...	/src/img/Input/P...	Pascasejana UI...	Hasil Telusur.Jl. I...	0.0864418	0.0864418	47

Gambar 4.5 *Form* pengujian

Form pengujian akan menampilkan hasil pengujian dari setiap pengenalan citra yang dilakukan di *form* cari. Setiap hasil pengenalan citra akan di hitung *precision* dan *recall*nya yang kemudian akan disimpan hasil dari proses ke dalam *database*. Pada bagian atas kanan *form* pengujian terdapat tombol yang berfungsi untuk mengubah data pengujian pada *form* ini kedalam format *Microsoft Excel*.

Program Pengenalan Tempat

Cari Tampil Pengujian Tentang Keluar

	Nama : Mey Hendra Putra Sirat NIM : 0701163068 Tempat Tanggal Lahir : Tanjungbalai 21 mei 1998 Fakultas : Sains dan Teknologi Jurusan : Ilmu Komputer Kampus : Universitas Islam Negeri Sumatera Utara
---	---

Status


Windows taskbar: 10:50, 29/07/2020

Gambar 4.6 *Form* Tentang

Pada *form* tentang adalah biodata dari hasil yang mengajukan judul tersebut yang terdapat diantaranya foto, nama, nim, tempat tanggal lahir, fakultas, jurusan, dan kampus.

Program Pengenalan Tempat

Cari Tampil Pengujian Tentang Keluar

	Nama : Mey Hendra Putra Sirat NIM : 0701163068 Tempat Tanggal Lahir : Tanjungbalai 21 mei 1998 Fakultas : Sains dan Teknologi Jurusan : Ilmu Komputer Kampus : Universitas Islam Negeri Sumatera Utara
---	---

Exit

Yakin keluar?

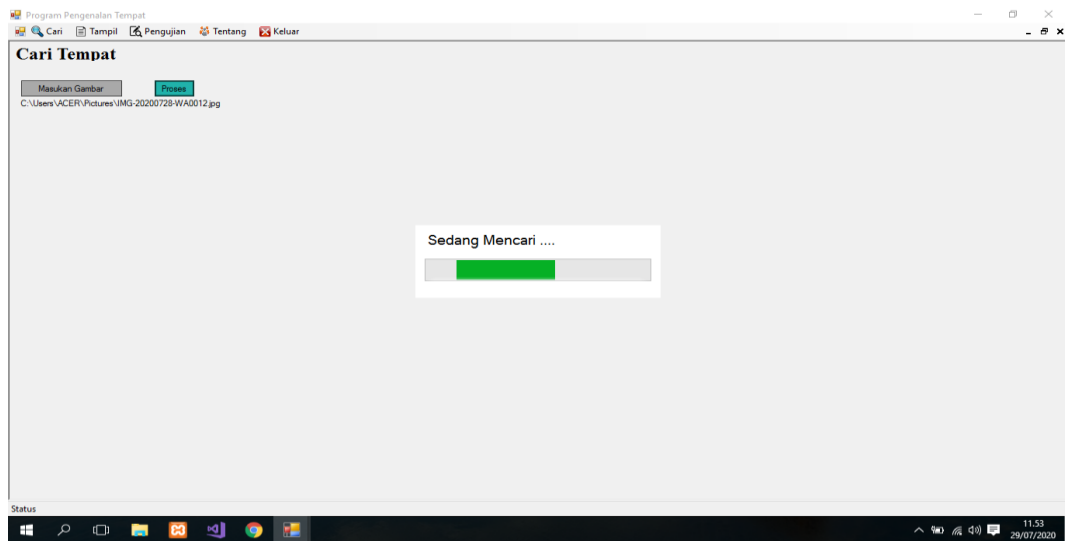
Yes No

Status

Windows taskbar: 10:50, 29/07/2020

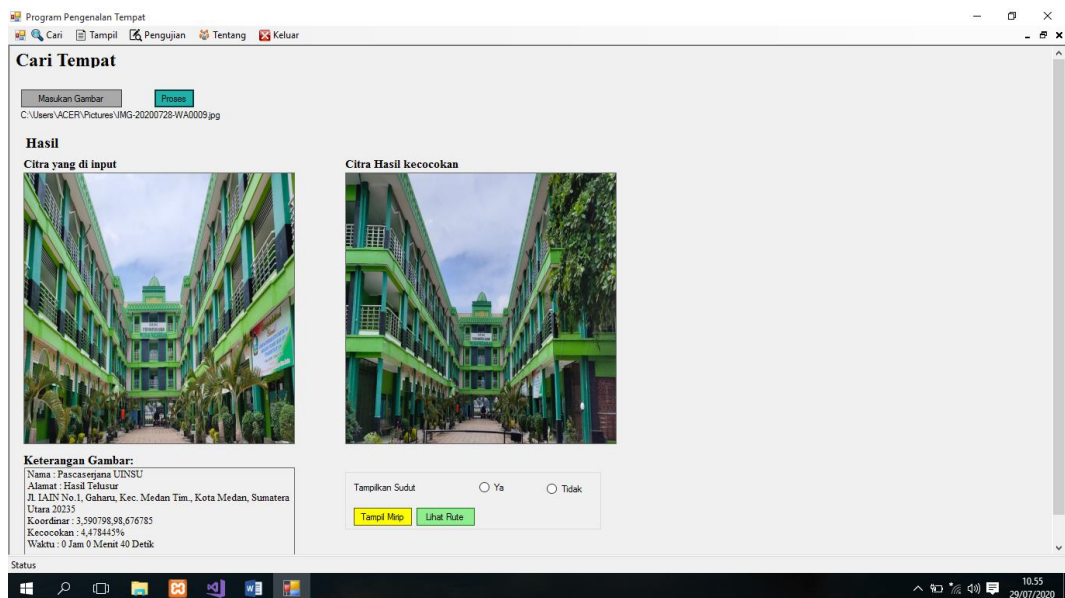
Gambar 4.7 *Form* keluar

Form ini bertujuan untuk keluar jika proses pencarian sudah selesai yang dilakukan pada *user*.



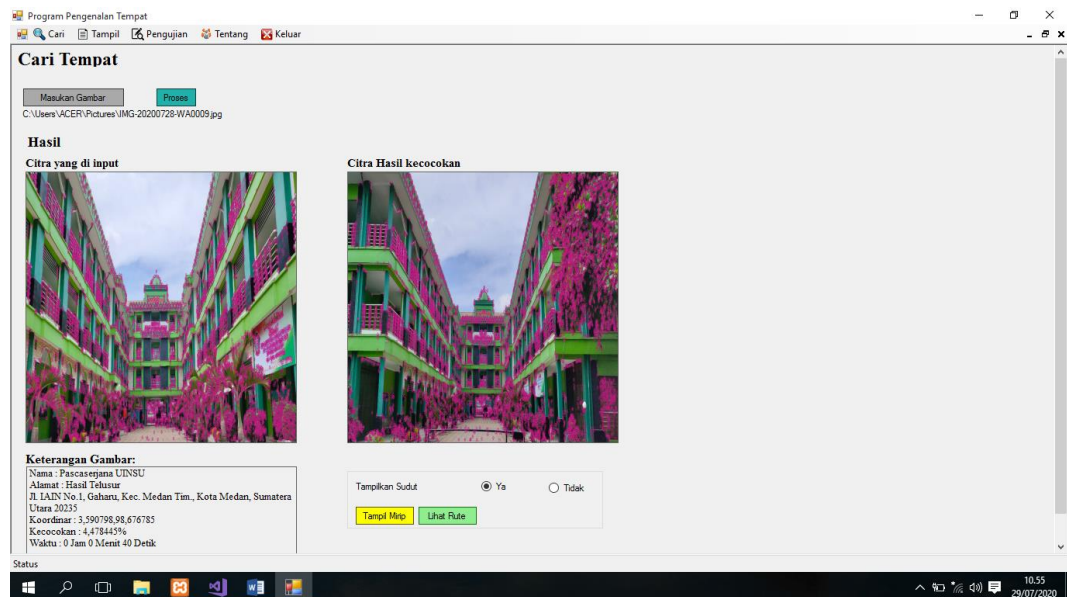
Gambar 4.9 Proses pencarian yang di *input*

Tunggu beberapa menit agar program dapat memproses pencocokan citra yang telah di *input*.



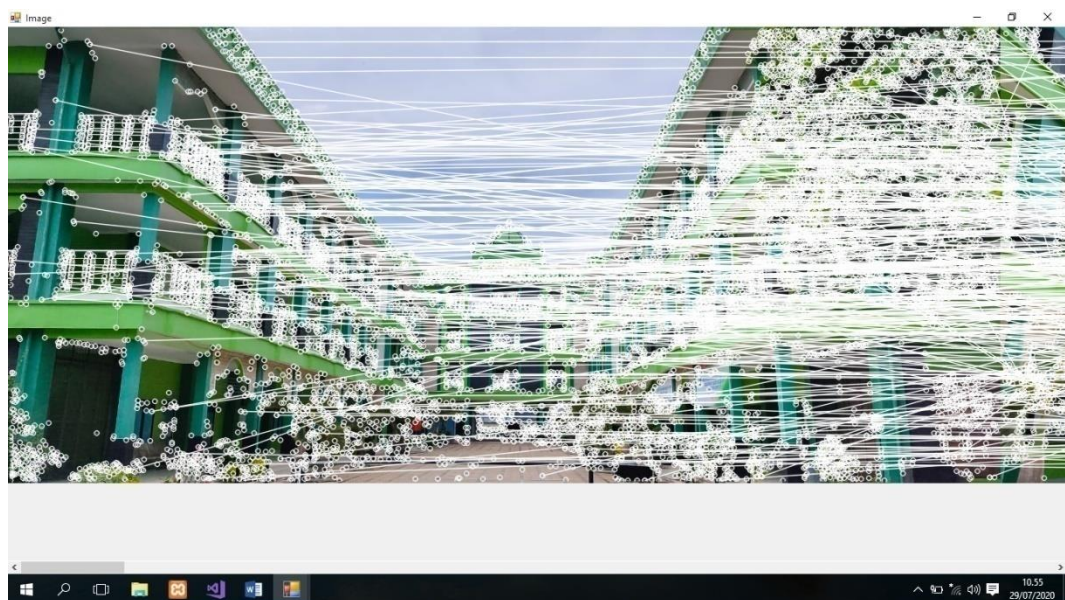
Gambar 4.10 Hasil dari penocokan citra

Pada gambar 4.10 adalah hasil dari proses yang dilakukan untuk pencocokan citra yang akan dilakukan untuk pengenalan tempat.



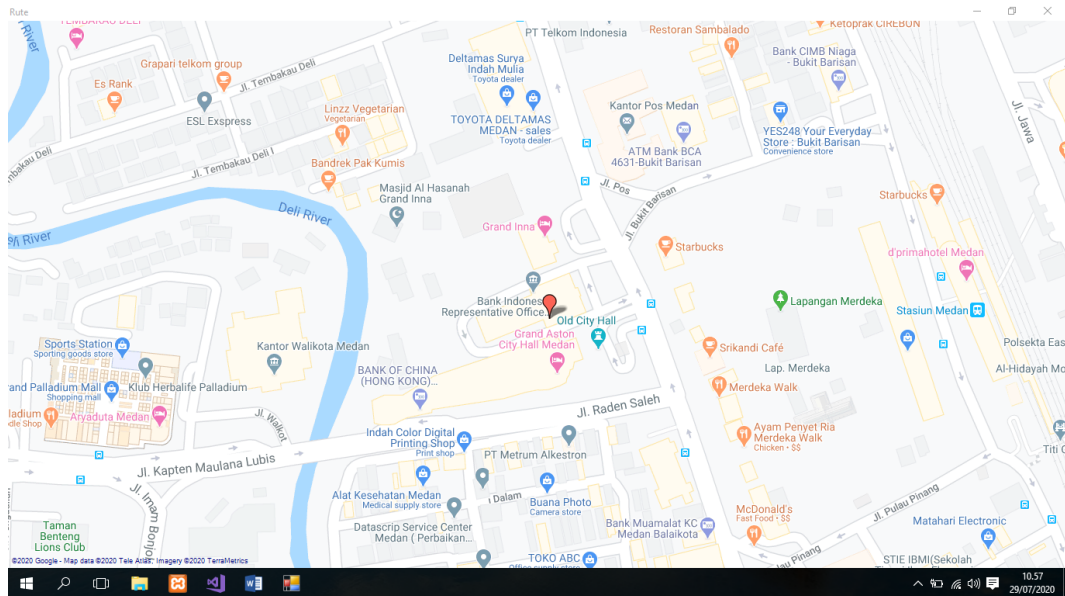
Gambar 4.11 Tampilan pencocokan sudut pada citra

Untuk gambar 4.11 adalah hasil dari pencocokan citra yang di tampilkan dari sudut kemiripan citra tersebut.



Gambar 4.12 Tampilan mirip pada citra

Dan yang terakhir adalah menampilkan *route* atau lokasi pada gambar yang telah di *input* untuk mencari kecocokan citra tersebut seperti pada gambar 4.13.



Gambar 4.13 *Route* lokasi

4.2.2 Penerapan

Penerapan dalam penggunaan sistem program ini adalah dapat mengetahui dimana kecocokan pada citra yang ingin diketahui serta dapat membantu seseorang yang ingin tau letak lokasi yang terdapat pada citra tersebut.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada penelitian kali ini dengan menggunakan metode SURF dan FAST dapat disimpulkan dalam pengenalan tempat yaitu :

Pengenalan berhasil pada citra dengan warna, dan resolusi yang berbeda. Namun bisa aja akan terjadinya kegagalan pada pengenalan dari pengujian diatas jika tempat yang tersimpan di *database* lebih dari sekitaran 200 data. Hal ini dikarenakan dalam melakukan pencocokan bergantung pada *precision* dan *recall* yang menyatakan kemiripan citra. *Precision* dan *Recall* yang dihasilkan pada citra yang gagal lebih tinggi jika dibandingkan dengan citra yang asli sehingga hasil pengenalan menjadi berubah dan menyebabkan kegagalan dalam pengenalan.

Dari pengujian yang dilakukan keberhasilan masih tetap berhasil dengan nilai *precision* 0,9. Yang berarti nilai kemiripan dari hasil pencocokan adalah 9% dan dengan nilai *recall* mencapai 100%. Nilai *recall* 100% berarti bahwa jumlah titik yang mirip adalah sama dengan jumlah titik yang di cocokkan.

5.2 Saran

Pada penelitian kali ini penulis menyatakan bahwasanya penelitian ini tidak sepenuhnya sempurna masih banyak kekurangan yang terdapat dari proses penelitian mau pun dalam proses pengembangan program yang di ciptakan. Adapun beberapa saran pada program kali ini yaitu keakuratan terhadap citra yang sama namun warna yang berbeda dapat menurun drastis dan masih ada pengenalan yang gagal baik citra warna, citra *grayscale*, citra *black white* atau citra yang terpotong karena nilai *precision* dan *recall* pada program yang dihasilkan dari pencocokan dengan citra lain lebih tinggi sehingga akan lebih baik apabila pengembangan selanjutnya dapat meningkatkan keakuratan dari pengenalan yang dilakukan.

DAFTAR PUSTAKA

- Andono, Pulung Nurtantio., dkk., 2017, *Pengolahan Citra Digital*, Yogyakarta : CV Andi
- Aditya Rizki Yudiantika, Selo Sulistyo, Bimo Sunarfri Hantono, 2014, Evaluasi Metode Pelacakan tanpa *Marker* pada *Metaio* SDK untuk pengembangan aplikasi kuis berbasis *Augmented Reality* di Museum, Jurnal Teknik Elektro dan Teknik Informasi Universitas Gajah Mada, ISSN 2302-3805. 1-6.
- Adelia,Jimmy Setiawan, 2011, Implementasi Customer Relationship Management (CRM) pada Sistem Reservasi Hotel Berbasis Website dan Desktop, Jurnal Teknik Informasi, September, 6(2) : 113-208
- Basuki, A. 2006. *Grafika Komputer: Teori dan Implementasi*. Yogyakarta:Andi.
- Bay, H., Tinne, T. & Luc, v. G., 2006. *SURF : Speeded Up Robust Features*. 1-14.
- Edouard Oyallon, J. R., 2015. An Analysis of the SURF Method., pp. 43-176., ISSN 2105-1232. 5: 177-218
- Faizal Zuli, 2018,*Rancangan Bangun Augmented dan Virtual Reality Menggunakan Algoritma FAST Sebagai Media Informasi 3D di Universitas Satya Negara Indonesia*, Jurnal Algoritma , Logika dan Komputasi, e-ISSN 2621-9840 | p-ISSN 2620 – 620X. 1(2): 94-104
- Firma Firmansyah Adi, Muhammad Ichwan, Yusup Miftahuddin, 2017, *Implementasi Algoritam Speed Up Robust Features (SURF) Pada Pengenalan Rambu-rambu Lalu Lintas*, Jurnal Teknik Informatika dan Sistem Informasi, Desember,e-ISSN 2443-2229. 3(3): 575-587
- Felix Pidha Hilman, 2015, *Perbandingan Metode SURF dan Sifat dalam Sistem Indentifikasi Tanda Tangan*, Jurnal Teknik Telekomunikasi, Agustus, e-ISSN 2355-9465. 2(2): 2467-2481
- Furqan,Muhammad,dkk, 2009, Smooth Support Vector Mechanie For Face Recognition Using Principal Component Analysis, ISSN 1978-5933, 293-298
- Furqan,Muhammad,dkk, 2020, Penerapan Metode Otsu dalam Melakukan Segmentasi Citra pada Naskah Arab, Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer, November, ISSN 2476-9843. 20(1) : 59-72

- Johar Nur Lin, M.Niswar, Ardiaty Arief, 2017, *Sistem Deteksi dan Ekstraksi Keypoint dari Bagian Jenis Sayap Nyamuk Dengan Menggunakan Algoritma Speed Up Robust Features*, Jurnal Sains dan Teknologi, Desember, ISSN 2303-3614. 6(2): 190-196
- Kurniawan Rakhmat, 2017 Rancangan Bangunan Aplikasi Pengaman Isi File Dokumen Dengan Algoritma RSA, Jurnal Ilmu Komputer, November, ISSN 2598-6341. 1(1) : 46-52
- Leon, S. J., 1998. *Ed. 5 th . Linear Algebra With Application*. New Jersey: Prentice Hall.
- Munir, R., 2004. *Pengolahan citra digital dengan pendekatan algoritmik*. Bandung: Informatika.
- Murni, A. A. & Setiawan, S., 1992. *Pengantar Pengolahan Citra Digital*. Jakarta: PT Elex Media Komputindo.
- Muhammad Baresi Ariel, Ratri Dwi Atmaja, Azizah Azizah, 2016, Implementasi Metode *Speed Up Robust Feature* dan *Scale Invariant Feature Transform* untuk Identifikasi Telapak Kaki Individu, Jurnal Sains dan Teknologi, p-ISSN 2087-9725, e-ISSN 2355-8059. 3(4) : 178-186
- Nasution, Yusuf Ramadhan, 2017 Menentukan Tingkat Kemacetan Lalulintas Dengan Metode Analytical Hierarchy Process(AHP), Jurnal Ilmu Komputer, November, ISSN 2598-6341. 1(1) : 40-45
- Ninuk Wiliani, Syadid Zambi, 2017 Rancangan Bangunan Aplikasi Kasir Tiket Nonton Bola Bareng pada X Kasir Disuatu Lokasi X dengan Visual Basic 2010 dan Mysql, Jurnal Rekayasa Informasi, Oktober, ISSN 2252-7354, 6(2) : 77-83
- Putra, Darma. 2010. *Pengolah Citra Digital*. Yogyakarta: CV Andi.
- Powers, D. M. W., 2007. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation*. 1-24.
- Pairuran T, Toban. 2016. *Pengolahan Citra*. Manado : Polimdo Press
- Rastri Parthivi, Vincent Suhartono, Guruh Fajar Shidik, 2015, *Feature Recognition Berbasis Corner Detecition Dengan Metode FAST,SURF,dan Flaan Tree Untuk Identifikasi Logo pada Augmented Reality Mobile System*, Jurnal Teknologi Informasi, Oktober,ISSN 1414-9999. 11(2): 100-15

Rosten & Drummond, 2006. *Machine learning for high speed corner detection*. 1-15

Sriani,dkk, 2017, Pendekomposisian Citra Digital Dengan Algoritma DWT, Jurnal Ilmu komputer, November, ISSN 2598-6341. 1(1) : 35-39

Sutoyo, et al., 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Andi.

PSEUDOCODE (PROGRAM)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace Place_Recognition
{
    public class clsfunction
    {
        public MySqlConnection OpenDB()
        {
            string constr = "Data Source=localhost; Initial Catalog = dbta; User ID = root; Password = ";

            MySqlConnection db = new MySqlConnection(constr);

            db.Open();

            return db;
        }

        public void CloseDB(MySqlConnection db)
        {
            db.Close();
        }

        public List<string> GetAllDtTmpt()
        {
            List<string> arr = new List<string>();

            MySqlConnection db = this.OpenDB();

            string query = "SELECT COUNT(ID) FROM dttmpt";

            MySqlCommand sql = new MySqlCommand(query, db);
            MySqlDataReader result = sql.ExecuteReader();
            result.Read();

            if (Convert.ToInt32(result[0]) > 0)
            {
                query = "SELECT ID FROM dttmpt ORDER BY CAST(ID AS UNSIGNED)";

                result.Close();
                sql = new MySqlCommand(query, db);
                result = sql.ExecuteReader();

                while (result.Read())
```

```

arr.Add(result[0].ToString());
    }

this.CloseDB(db);

return arr;
}

public List<string> GetAllUji()
{
    List<string> arr = new List<string>();

    MySqlConnection db = this.OpenDB();

    string query = "SELECT COUNT(ID) FROM dthuji";

    MySqlCommand sql = new MySqlCommand(query, db);
    MySqlDataReader result = sql.ExecuteReader();
    result.Read();

    if (Convert.ToInt32(result[0]) > 0)
    {
        query = "SELECT ID FROM dthuji ORDER BY ID";

        result.Close();
        sql = new MySqlCommand(query, db);
        result = sql.ExecuteReader();

        while (result.Read())
            arr.Add(result[0].ToString());
    }

    this.CloseDB(db);

    return arr;
}

public string GetGoogleApi()
{
    return "AIzaSyBgtClvyeRv6JDfVMKtgx7PKRToFetneCs";
}
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Drawing;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Util;
using Emgu.CV.Cuda;

```

```

using Emgu.CV.XFeatures2D;

namespace Place_Recognition
{
    public class Citra
    {
        public Image<Bgr, byte> img;
        public Image<Gray, byte> imggrey;
        public string link;

        public Citra(string link)
        {
            this.link = link;
            this.img = new Image<Bgr, byte>(link);

            this.imggrey = this.Grayscale();
        }

        public Image<Gray, byte> Grayscale()
        {
            Matrix<byte> x = new Matrix<byte>(this.img.Mat.Size);

            for (int i = 0; i < this.img.Rows; i++)
            {
                for (int j = 0; j < this.img.Cols; j++)
                {
                    x[i, j] = Convert.ToByte(0.114 * this.img.Data[i, j, 0] +
                    0.587 * this.img.Data[i, j, 1] + 0.299 * this.img.Data[i, j, 2]);
                }
            }

            Mat y = new Mat();
            x.Mat.ConvertTo(y, DepthType.Cv64F);

            return y.ToImage<Gray, byte>();
        }

        public Image<Gray, byte> Histogram(Image<Gray, byte> img)
        {
            Image<Gray, byte> x = new Image<Gray, byte>(img.Size);

            CvInvoke.CLAHE(img, 4, new Size(8, 8), x);

            return x;
        }

        public double GetMaxVal()
        {
            double x = 0;

            for (int i = 0; i < this.imggrey.Rows; i++)
            {
                for (int j = 0; j < this.imggrey.Cols; j++)
                {
                    if (this.imggrey[i, j].Intensity > x)
                    {
                        x = this.imggrey[i, j].Intensity;
                    }
                }
            }
        }
    }
}

```

```

return x;
    }
public VectorOfKeyPoint GetSudut(int thres)
{
    VectorOfKeyPoint key = new VectorOfKeyPoint();
    List<List<int>> x = newList<List<int>>();

    Image<Gray, byte> y = this.imggrey;

    int k = 0;

    for (int i = 3; i < this.imggrey.Rows - 3; i += 6)
    {
        for (int j = 3; j < this.imggrey.Cols - 3; j++)
        {
            int n = 0, n2 = 0, m = 0;

            int lp = Convert.ToInt32(y[i, j].Intensity);
            int p1 = Convert.ToInt32(y[i - 3, j].Intensity);
            int p2 = Convert.ToInt32(y[i - 3, j + 1].Intensity);
            int p3 = Convert.ToInt32(y[i - 2, j + 2].Intensity);
            int p4 = Convert.ToInt32(y[i - 1, j + 3].Intensity);
            int p5 = Convert.ToInt32(y[i, j + 3].Intensity);
            int p6 = Convert.ToInt32(y[i + 1, j + 3].Intensity);
            int p7 = Convert.ToInt32(y[i + 2, j + 2].Intensity);
            int p8 = Convert.ToInt32(y[i + 3, j + 1].Intensity);
            int p9 = Convert.ToInt32(y[i + 3, j].Intensity);
            int p10 = Convert.ToInt32(y[i + 3, j - 1].Intensity);
            int p11 = Convert.ToInt32(y[i + 2, j - 2].Intensity);
            int p12 = Convert.ToInt32(y[i + 1, j - 3].Intensity);
            int p13 = Convert.ToInt32(y[i, j - 3].Intensity);
            int p14 = Convert.ToInt32(y[i - 1, j - 3].Intensity);
            int p15 = Convert.ToInt32(y[i - 2, j - 2].Intensity);
            int p16 = Convert.ToInt32(y[i - 3, j - 1].Intensity);

            if (lp + thres > p1)
                n++;

            if (lp + thres > p5)
                n++;

            if (lp + thres > p9)
                n++;

            if (lp + thres > p13)
                n++;

            if (n < 3)
            {
                if (lp - thres < p1)
                    n2++;

                if (lp - thres < p5)
                    n2++;

                if (lp - thres < p9)

```

```

n2++;

if (lp - thres< p13)
    n2++;
}

if (n >= 3)
{
if (lp + thres> p1)
    m++;

if (lp + thres> p2)
    m++;

if (lp + thres> p3)
    m++;

if (lp + thres> p4)
    m++;

if (lp + thres> p5)
    m++;

if (lp + thres> p6)
    m++;

if (lp + thres> p7)
    m++;

if (lp + thres> p8)
    m++;

if (lp + thres> p9)
    m++;

if (lp + thres> p10)
    m++;

if (lp + thres> p11)
    m++;

if (lp + thres> p12)
    m++;

if (lp + thres> p13)
    m++;

if (lp + thres> p14)
    m++;

if (lp + thres> p15)
    m++;

if (lp + thres> p16)
    m++;
}

```

```

if (n2 >= 3)
{
if (lp - thres> p1)
m++;

if (lp - thres> p2)
m++;

if (lp - thres> p3)
m++;

if (lp - thres> p4)
m++;

if (lp - thres> p5)
m++;

if (lp - thres> p6)
m++;

if (lp - thres> p7)
m++;

if (lp - thres> p8)
m++;

if (lp - thres> p9)
m++;

if (lp - thres> p10)
m++;

if (lp - thres> p11)
m++;

if (lp - thres> p12)
m++;

if (lp - thres> p13)
m++;

if (lp - thres> p14)
m++;

if (lp - thres> p15)
m++;

if (lp - thres> p16)
m++;
}

if (m >= 9)
{
int score = Math.Abs(lp - p1) + Math.Abs(lp - p2) + Math.Abs(lp - p3) +
Math.Abs(lp - p4) + Math.Abs(lp - p5) + Math.Abs(lp - p6) + Math.Abs(lp -
p7) + Math.Abs(lp - p8) + Math.Abs(lp - p9) + Math.Abs(lp - p10) +

```

```

Math.Abs(lp - p11) + Math.Abs(lp - p12) + Math.Abs(lp - p13) + Math.Abs(lp -
p14) + Math.Abs(lp - p15) + Math.Abs(lp - p16);
x.Add(newList<int>());
        x[k].Add(i);
        x[k].Add(j);
        x[k].Add(score);

        j += 9;

        k++;
    }
}

for (inti = 1; i<x.Count; i++)
{
    if (AreAdj(x[i-1][0], x[i-1][1], x[i][0], x[i][1]))
    {
        if (x[i][2] > x[i - 1][2])
        x.RemoveAt(i - 1);
        else
        x.RemoveAt(i);
    }
}

MKeyPoint[] kp = newMKeyPoint[x.Count];
for (inti = 0; i<x.Count; i++)
{
    MKeyPoint res = newMKeyPoint();
    res.Size = 7;
    res.Octave = 0;
    res.Angle = -1;
    res.Point = newPointF(x[i][1], x[i][0]);
    kp[i] = res;
}

key = newVectorOfKeyPoint(kp);

return key;
}

publicboolAreAdj(int x, int y, int x2, int y2)
{
    intxDist = x - x2;
    intyDist = y - y2;

    returnMath.Sqrt(Math.Pow(xDist,2) + Math.Pow(yDist,2)) <= 10;
}

publicfloatGetHaarx(float x, float y, float lambda)
{
    return -(this.Konvolusi(1, -lambda - 1, -lambda - 1, 2 * lambda + 1, x, y) +
this.Konvolusi(0, -lambda - 1, lambda + 1, 2 * lambda + 1, x, y));
}

publicfloatGetHaary(float x, float y, float lambda)

```



```

        {
return -(this.Konvolusi(-lambda - 1, 1, 2 * lambda + 1, -lambda - 1, x, y) +
this.Konvolusi(-lambda - 1, 0, 2 * lambda + 1, lambda + 1, x, y));
        }

public float Konvolusi(float a, float b, float c, float d, float x, float y)
{
Image<Gray, byte>img = this.imggrey;

int a1 = Convert.ToInt32(Math.Abs(x - a));
int a2 = Convert.ToInt32(Math.Abs(y - b));
int b1 = Convert.ToInt32(Math.Abs(a1 - c));
int b2 = Convert.ToInt32(Math.Abs(a2 - d));

if (a1 >= img.Height)
    a1 = img.Height - 1;

if (b1 >= img.Height)
    b1 = img.Height - 1;

if (a2 >= img.Width)
    a2 = img.Width - 1;

if (b2 >= img.Width)
    b2 = img.Width - 1;

return Convert.ToInt32(img[b1, b2].Intensity) + Convert.ToInt32(img[a1,
a2].Intensity) - Convert.ToInt32(img[b1, a2].Intensity) -
Convert.ToInt32(img[a1, b2].Intensity);
}

public Mat GetDeskriptor(int thres)
{
VectorOfKeyPoint key = this.GetSudut(thres);
Matrix<float> mat = new Matrix<float>(key.Size, 64);

for (int h = 0; h < key.Size; h++)
{
double skala = key[h].Size * 0.4, orientasi = key[h].Angle;

float cp = (float) Math.Cos(orientasi), sp = (float) Math.Sin(orientasi);

List<List<float>> des = newList<List<float>>();

for (int i = 0; i < 20; i++)
{
des.Add(newList<float>());
}

double norms = 0;
for (int i = 0; i <= 3; i++)
{
for (int j = 0; j <= 3; j++)
{
int idx = 4 * i + j;

des[idx].Add(0);

```

```

        des[idx].Add(0);
        des[idx].Add(0);
        des[idx].Add(0);

    for(int k = 0; k <= 4; k++)
    {
    for(int l = 0; l <= 4; l++)
    {
        float u = Convert.ToSingle(key[h].Point.Y + skala * (cp * ((i - 2) * 5 + k + 0.5) - sp * ((j - 2) * 5 + l + 0.5)));
        float v = Convert.ToSingle(key[h].Point.X + skala * (sp * ((i - 2) * 5 + k + 0.5) + cp * ((j - 2) * 5 + l + 0.5)));

        float responx = Convert.ToSingle(this.GetHaarx(u, v, Convert.ToInt32(skala + 0.5)));
        float respony = Convert.ToSingle(this.GetHaary(u, v, Convert.ToInt32(skala + 0.5)));

        float gauss = Convert.ToSingle(1 / (2 * Math.PI * Math.Pow(3.3, 2)) * Math.Exp(-(Math.Pow((i-2)*5+k+0.5, 2) + Math.Pow((j-2)*5+l+0.5),2)) / (2 * Math.Pow(3.3, 2))));

        float responu = Convert.ToSingle(gauss * (responx * cp + respony * sp));
        float responv = Convert.ToSingle(gauss * (-responx * sp + respony * cp));

        des[idx][0] += Convert.ToSingle(responu);
        des[idx][1] += Convert.ToSingle(Math.Abs(responu));
        des[idx][2] += Convert.ToSingle(responv);
        des[idx][3] += Convert.ToSingle(Math.Abs(responv));
    }
    }

    norms += Math.Pow(des[idx][0], 2) + Math.Pow(des[idx][1], 2) + Math.Pow(des[idx][2], 2) + Math.Pow(des[idx][3], 2);
    }

    norms = (float)Math.Sqrt(norms);

    if(norms != 0)
    {
    for(int i = 0; i < 16; i++)
    {
        des[i][0] /= (float)norms;
        des[i][1] /= (float)norms;
        des[i][2] /= (float)norms;
        des[i][3] /= (float)norms;
    }
    }

    int y = 0;
    for(int i = 0; i < 16; i++)
    {
    for(int j = 0; j < 4; j++)

```

```

        {
            mat[h, y] = des[i][j];
            y++;
        }
    }
}

Mat ndes = new Mat();
mat.Mat.ConvertTo(ndes, DepthType.Cv32F);

return ndes;
}

public Image<Bgr, byte> GambarSudut(int thres)
{
    Mat result = new Mat();

    Features2DToolbox.DrawKeypoints(this.img, this.GetSudut(thres), result,
    new Bgr(Color.MediumVioletRed));

    return result.ToImage<Bgr, byte>();
}

}

Image<Bgr, byte> GambarSudut(int thres)
{
    Mat result = new Mat();

    Features2DToolbox.DrawKeypoints(this.img, this.GetSudut(thres), result,
    new Bgr(Color.MediumVioletRed));

    return result.ToImage<Bgr, byte>();
}

}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace Place_Recognition
{
    public class Tempat
    {
        public string id, nama, src, lat, lng, almt, desc;
        public int row, col;
        public Citra citra;
        private clsfunction clsfunc;

        public Tempat(string x)
        {
            this.id = x;

```

```

this.clsfunc = newclsfunction();
    }

publicvoid SetDataDB()
    {
        MySqlConnection db = this.clsfunc.OpenDB();

        string query = "SELECT NAMA, LINK, LAT, LNG, ALAMAT FROM dttmpt "+
            "WHERE ID = '"+this.id+"'";

        MySqlCommand sql = new MySqlCommand(query, db);
        MySqlDataReader result = sql.ExecuteReader();
        result.Read();

        this.nama = result[0].ToString();
        this.src = result[1].ToString();
        this.lat = result[2].ToString();
        this.lng = result[3].ToString();
        this.almt = result[4].ToString();
        this.citra = new Citra(this.src);

        this.clsfunc.CloseDB(db);
    }

publicvoid SetDataDB2()
    {
        MySqlConnection db = this.clsfunc.OpenDB();

        string query = "SELECT NAMA, LINK, LAT, LNG, ALAMAT, DESCRIP, ROW, COL FROM
            dttmpt "+
            "WHERE ID = '"+this.id+"'";

        MySqlCommand sql = new MySqlCommand(query, db);
        MySqlDataReader result = sql.ExecuteReader();
        result.Read();

        this.nama = result[0].ToString();
        this.src = result[1].ToString();
        this.lat = result[2].ToString();
        this.lng = result[3].ToString();
        this.almt = result[4].ToString();
        this.desc = result[5].ToString();
        this.row = Convert.ToInt32(result[6]);
        this.col = Convert.ToInt32(result[7]);
        this.citra = new Citra(this.src);

        this.clsfunc.CloseDB(db);
    }

publicvoid UpdUji(string desc, int m, int n)
    {
        MySqlConnection db = this.clsfunc.OpenDB();

        string query = "UPDATE dttmpt2 SET DESCRIPTOR = '" + desc + "', ROW = '" + m + "' +
            "WHERE ID = '" + this.id + "'";

        MySqlCommand sql = new MySqlCommand(query, db);

```

```

MySQLDataReader result = sql.ExecuteReader();

this.clsfunc.CloseDB(db);
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Util;
using Emgu.CV.Cuda;
using Emgu.CV.XFeatures2D;
using System.Data.SqlClient;
using MySql.Data;
using MySql.Data.MySqlClient;

namespace Place_Recognition
{
    public class Pengujian
    {
        public string id;
        public float precision, recall;
        public string limg1, limg2;
        public double time;
        private clsfunction clsfunc;

        public Pengujian(string img1, string img2)
        {
            this.limg1 = img1;
            this.limg2 = img2;
            this.clsfunc = new clsfunction();
        }

        public Pengujian(string id)
        {
            this.id = id;
            this.clsfunc = new clsfunction();
        }

        public void SetDataDB()
        {
            MySqlConnection db = this.clsfunc.OpenDB();

            string query = "SELECT PREC, RECALL, WKT FROM dthuji" +
                "WHERE IMG1 = '" + this.limg1 + "' && IMG2 = '" + this.limg2 + "'";

            MySqlCommand sql = new MySqlCommand(query, db);
            MySQLDataReader result = sql.ExecuteReader();
            result.Read();

```

```

this.precision = Convert.ToSingle(result[0]);
this.recall = Convert.ToSingle(result[1]);
this.time = Convert.ToInt32(result[2]);

result.Close();

this.clsfunc.CloseDB(db);
    }

publicvoid SetDataDB2()
    {
        MySqlConnectiondb = this.clsfunc.OpenDB();

        string query = "SELECT IMG1, IMG2, PREC, RECALL, WKT FROM dthuji " +
            "WHERE ID = '" + this.id + "'";

        MySqlCommandsql = newMySqlCommand(query, db);
        MySqlDataReader result = sql.ExecuteReader();

        if (result.Read())
        {
            this.limg1 = result[0].ToString();
            this.limg2 = result[1].ToString();
            this.precision = Convert.ToSingle(result[2]);
            this.recall = Convert.ToSingle(result[3]);
            this.time = Convert.ToInt32(result[4]);
        }

        result.Close();

        this.clsfunc.CloseDB(db);
    }

publicstringGetID()
    {
        return"UJI-"+this.CountID();
    }

publicstringCountID()
    {
        stringhsl = "";

        MySqlConnectiondb = this.clsfunc.OpenDB();

        string query = "SELECT ID FROM dthuji ORDER BY ID DESC";

        MySqlCommandsql = newMySqlCommand(query, db);
        MySqlDataReader result = sql.ExecuteReader();

        if (result.Read())
        {
            string[] x = result[0].ToString().Split('-');

            hsl = (Convert.ToInt32(x[1]) + 1).ToString();
        }
        else
    }

```

```

        hsl = "1";

        return this.SetLenID(hsl);
    }

    public string SetLenID(string x)
    {
        if (x.Length == 1)
            return "000" + x;
        elseif (x.Length == 2)
            return "00" + x;
        elseif (x.Length == 3)
            return "0" + x;

        return x;
    }

    public void NewUji()
    {
        MySqlConnection db = this.clsfunc.OpenDB();

        string query = "INSERT INTO dthuji" +
            "(ID, IMG1, IMG2, PREC, RECALL, WKT)" +
            "VALUES" +
            "(" + this.GetID() + "', '" + this.limg1 + "', '" + this.limg2 + "', '" +
            this.precision.ToString().Replace(',', '.') + "', '" +
            this.recall.ToString().Replace(',', '.') + "', '" + this.time + "')";

        MySqlCommand sql = new MySqlCommand(query, db);
        MySqlDataReader result = sql.ExecuteReader();
        result.Read();
        result.Close();

        this.clsfunc.CloseDB(db);
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Diagnostics;
using System.Text;
using System.Threading.Tasks;
using System.Threading;
using System.Windows.Forms;
using System.IO;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Util;
using Emgu.CV.Cuda;
using Emgu.CV.XFeatures2D;

namespace Place_Recognition

```

```

{
public class Cari
    {
public Citra img;
public Tempat img2;
public int threshold, k;
public TimeSpan wkt;
public Pengujian uji;
public double uthreshold;
public float cocok;
public clsfunction clsfunc;
public Mat mask, mask2;
public VectorOfVectorOfDMatch match;

public Cari(string link, int thres, int k)
    {
this.img = new Citra(link);
this.threshold = thres;
this.k = k;
this.uthreshold = 0.8;
this.cocok = 0;
this.clsfunc = new clsfunction();
this.mask = new Mat();
this.mask2 = new Mat();
this.match = new VectorOfVectorOfDMatch();
    }
public float GetRecall(Mat mask, VectorOfKeyPoint key)
    {
return Convert.ToSingle(this.CMatch(mask)) / Convert.ToSingle(key.Size);
    }
public float GetPrec(Mat mask)
    {
return Convert.ToSingle(this.CMatch(mask)) /
Convert.ToSingle(this.AllMatch(mask));
    }
public void Match()
    {
Stopwatch watch;
        watch = Stopwatch.StartNew();

Mat mask, des1, des2, mask2;
VectorOfKeyPoint key1, key2;

        des1 = this.img.GetDescriptor(this.threshold);

this.img.imggrey = this.img.Histogram(this.img.imggrey);

List<string> lst = this.clsfunc.GetAllDtTmp();

for (int i = 0; i < lst.Count(); i++)
    {
List<List<int>>> lstmd = newList<List<int>>>();
VectorOfVectorOfDMatch matches = new VectorOfVectorOfDMatch();
Tempat tmp = new Tempat(lst[i]);
tmp.SetDataDB();
tmp.citra.imggrey = tmp.citra.Histogram(tmp.citra.imggrey);
        des2 = tmp.citra.GetDescriptor(this.threshold);
    }
}
}

```



```

BFMatcher matcher = newBFMatcher(DistanceType.L2);
matcher.Add(des1);

matcher.KnnMatch(des2, matches, k, null);

        mask = newMat(matches.Size, 1, DepthType.Cv8U, 1);
        mask2 = newMat(matches.Size, 1, DepthType.Cv8U, 1);
mask.SetTo(newMCvScalar(255));
        mask2.SetTo(newMCvScalar(255));

Features2DToolbox.VoteForUniqueness(matches, this.uthreshold, mask);
Features2DToolbox.VoteForUniqueness(matches, 1, mask2);

int nonZeroCount = CvInvoke.CountNonZero(mask);
if (nonZeroCount >= 4)
{
    if (this.GetPrec(mask, mask2, tmp, matches) > this.cocok)
    {
        this.img2 = newTempat(1st[i]);
        this.img2.SetDataDB();
        this.mask = newMat();
        this.mask = mask;
        this.mask2 = newMat();
        this.mask2 = mask2;
        this.match = newVectorOfVectorOfDMatch();
        this.match = matches;
        this.cocok = this.GetPrec(this.mask, this.mask2, this.img2, this.match);

        if (this.cocok >= 1)
            break;
    }
}

watch.Stop();
this.wkt = watch.Elapsed;
}

public int CMatch(VectorOfKeyPoint key1, VectorOfKeyPoint key2,
VectorOfVectorOfDMatch matches, Mat mask)
{
    return CvInvoke.CountNonZero(mask);
}

public int AllMatch(Mat mask)
{
    Matrix<byte> nmask = newMatrix<byte>(mask.Rows, mask.Cols,
mask.NumberOfChannels);

    mask.CopyTo(nmask);

    return nmask.ManagedArray.OfType<byte>().ToList().Count();
}

public void MoveFile()
{
    string nlink = "./src/img/input/" + this.img2.nama + "-" + DateTime.Now.Year
+ "-" + DateTime.Now.Month + "-" + DateTime.Now.Day + " " +

```

```

DateTime.Now.Hour + "-" + DateTime.Now.Minute + "-" + DateTime.Now.Second +
Path.GetExtension(this.img.link);
File.Copy(this.img.link, nlink);

this.uji = newPengujian(this.img2.id, nlink);

this.uji.precision = this.cocok;
this.uji.recall = this.GetRecall();
this.uji.time = this.wkt.TotalSeconds;
this.uji.NewUji();
    }

publicMatDrawMatch()
    {
Mat result = newMat();

Features2DToolbox.DrawMatches(this.img2.citra.img,
this.img2.citra.GetSudut(this.threshold), this.img.img,
this.img.GetSudut(this.threshold), this.match, result, newMCvScalar(255,
255, 255), newMCvScalar(255, 255, 255), this.mask);

return result;
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Diagnostics;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Util;
using Emgu.CV.Cuda;
using Emgu.CV.XFeatures2D;
using System.Threading;

namespace Place_Recognition
{
    publicpartialclassMainFrm : Form
    {
        privateintchildFormNumber = 0;

        publicMainFrm()
        {
            InitializeComponent();
        }
    }
}

```

```

private void mnu_exit_Click(object sender, EventArgs e)
{
    DialogResult exit = MessageBox.Show("Yakin keluar?", "Exit",
    MessageBoxButtons.YesNo);

    if (exit == DialogResult.Yes)
    this.Close();
}

private void mnu_cari_Click(object sender, EventArgs e)
{
    this.lbl_main.Hide();

    CariFrm frm = new CariFrm();

    frm.MdiParent = this;
    frm.WindowState = FormWindowState.Maximized;
    frm.Show();
}

private void mnu_tampil_Click(object sender, EventArgs e)
{
    this.lbl_main.Hide();

    TampilFrm frm = new TampilFrm();

    frm.MdiParent = this;
    frm.WindowState = FormWindowState.Maximized;
    frm.Show();
}

private void mnu_uji_Click(object sender, EventArgs e)
{
    this.lbl_main.Hide();

    UjiFrm frm = new UjiFrm();

    frm.MdiParent = this;
    frm.WindowState = FormWindowState.Maximized;
    frm.Show();
}

private void mnu_ttg_Click(object sender, EventArgs e)
{
    this.lbl_main.Hide();

    TentangFrm frm = new TentangFrm();

    frm.MdiParent = this;
    frm.WindowState = FormWindowState.Maximized;
    frm.Show();
}
}

using System;
using System.Collections.Generic;

```

```

using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Util;
using Emgu.CV.Cuda;
using Emgu.CV.XFeatures2D;

namespace Place_Recognition
{
    public partial class CariFrm : Form
    {
        Caricari;
        public CariFrm()
        {
            InitializeComponent();
            this.WindowState = FormWindowState.Maximized;
        }

        private void CariFrm_Resize(object sender, EventArgs e)
        {
            this.WindowState = FormWindowState.Maximized;
        }

        private void btn_inpt_Click(object sender, EventArgs e)
        {
            OpenFileDialog open = new OpenFileDialog();
            open.Filter = "Image Files(*.jpg; *.jpeg; *.png)|*.jpg; *.jpeg; *.png";

            if (open.ShowDialog() == DialogResult.OK)
            {
                string[] filename = open.FileName.Split();
                lbl_nm_ctr.Text = open.FileName;

                this.cari = new Cari(open.FileName, 50, 2);

                this.UnTampil();

                btn_cari.Visible = true;
            }
        }

        private void Btn_cari_Click(object sender, EventArgs e)
        {
            using (ProcessFrm frm = new ProcessFrm(this.cari.Match))
            {
                frm.ShowDialog(this);
            }
        }
    }
}

```

```

this.cari.MoveFile();
this.Tampil();
}

private void Tampil()
{
this.pb_inpt.Image = new Bitmap(this.cari.img.link);
this.pb_hsl.Image = new Bitmap(this.cari.img2.citra.link);
this.lbl_isi_ket.Text = "Nama : " + this.cari.img2.nama + "\nAlamat : " +
this.cari.img2.almt + "\nKoordinat : " + this.cari.img2.lat + "," +
this.cari.img2.lng + "\nKecocokan : " + this.cari.cocok * 100 + "%\nWaktu :
" + this.cari.wkt.Hours.ToString() + " Jam " +
this.cari.wkt.Minutes.ToString() + " Menit " +
this.cari.wkt.Seconds.ToString() + " Detik";

this.pb_inpt.Visible = true;
this.pb_hsl.Visible = true;
this.gb_option.Visible = true;
this.lbl_ket.Visible = true;
this.lbl_hsl.Visible = true;
this.lbl_ctr_inpt.Visible = true;
this.lbl_ctr_hsl.Visible = true;
this.lbl_isi_ket.Visible = true;
this.btn_lht_rute.Visible = true;
this.btn_tmpl_mirip.Visible = true;
}

private void UnTampil()
{
this.pb_inpt.Visible = false;
this.pb_hsl.Visible = false;
this.gb_option.Visible = false;
this.lbl_ket.Visible = false;
this.lbl_hsl.Visible = false;
this.lbl_ctr_inpt.Visible = false;
this.lbl_ctr_hsl.Visible = false;
this.lbl_isi_ket.Visible = false;
this.btn_lht_rute.Visible = false;
this.btn_tmpl_mirip.Visible = false;
btn_cari.Visible = false;
}

private void Rdosdt_CheckedChanged(object sender, EventArgs e)
{
if (this.rdosdt.Checked)
{
this.pb_inpt.Image =
(this.cari.img.GambarSudut(this.cari.threshold).ToBitmap());
this.pb_hsl.Image =
(this.cari.img2.citra.GambarSudut(this.cari.threshold).ToBitmap());
}
}

private void Rdosdtn_CheckedChanged(object sender, EventArgs e)
{
if (this.rdosdtn.Checked)
{

```

```

        this.pb_inpt.Image = new Bitmap(this.cari.img.link);
        this.pb_hsl.Image = new Bitmap(this.cari.img2.citra.link);
    }
}

private void Btn_lht_rute_Click(object sender, EventArgs e)
{
    RuteFrm frm = new RuteFrm(Convert.ToString(this.cari.img2.lat),
    Convert.ToString(this.cari.img2.lng));

    frm.ShowDialog();
}

private void Btn_tmpl_mirip_Click(object sender, EventArgs e)
{
    Mat result = this.cari.DrawMatch();

    ImageViewer viewer = new ImageViewer(result);

    viewer.Size = new Size(result.Width, result.Height);

    viewer.ShowDialog();
}
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;
using System.Reflection;
using Emgu.CV;
using Emgu.CV.CvEnum;
using Emgu.CV.Features2D;
using Emgu.CV.Structure;
using Emgu.CV.UI;
using Emgu.CV.Util;
using Emgu.CV.Cuda;
using Emgu.CV.XFeatures2D;

namespace Place_Recognition
{
    public partial class TampilFrm : Form
    {
        private clsfunction clsfunc;
        private Tempattmpt;
        private List<string> lst = newList<string>();
        private int fst = 1, last, now = 1, gidx = 12;

        public TampilFrm()
        {
            InitializeComponent();

```

```

this.WindowState = FormWindowState.Maximized;

this.clsfunc = newclsfunction();

this.lst = this.clsfunc.GetAllDtTmpt();
this.lbllnk_prev.Enabled = false;
this.last = Convert.ToInt32(Math.Ceiling(Convert.ToDouble(lst.Count()) /
Convert.ToDouble(12)));
this.DataHal(1);
    }

privatevoidDataHal(int x)
    {
        x -= 1;
        intdri = x * 12, smpi = (x + 1) * 12;

        if (this.lbl_err.Visible)
            this.lbl_err.Visible = false;

        if (lst.Count() >dri)
        {
            Tampil(dri, smpi);

            if (smpi>lst.Count())
                UnTampil(smpi - lst.Count());
        }
        else
        {
            this.lbl_err.Visible = true;

            UnTampil(1);
        }
    }

privatevoidUnTampil(int x)
    {
        inti = x;

        stringidx = "", gb = "gb_tmpt";

        foreach (Control ctrl inthis.Controls)
        {
            if (ctrl isGroupBox)
            {
                if (i != 1)
                    idx = i.ToString();

                if (ctrl.Name == gb + idx)
                {
                    ctrl.Visible = false;
                }
            }
        }
    }
}

```

```

private void Lbl1lnk_prev_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now -= 1;
    this.DataHal(this.now);

    if (this.now == Convert.ToInt32(this.lbl1lnk_fst.Text))
        this.UpdLink(0);

    this.CekLink();
}

private void Lbl1lnk_next_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now += 1;
    this.DataHal(this.now);

    if (this.now == Convert.ToInt32(this.lbl1lnk_1st.Text))
        this.UpdLink(1);

    this.CekLink();
}

private void Lbl1lnk_1st_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now = Convert.ToInt32(this.lbl1lnk_1st.Text);

    this.DataHal(this.now);

    this.UpdLink(1);
    this.CekLink();
}

private void Lbl1lnk_bfr1st_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now = Convert.ToInt32(this.lbl1lnk_bfr1st.Text);

    this.DataHal(this.now);

    this.CekLink();
}

private void Lbl1lnk_fst_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now = Convert.ToInt32(this.lbl1lnk_fst.Text);

    this.DataHal(this.now);

    this.UpdLink(0);
    this.CekLink();
}

```



```

private void lblLnk_two_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now = Convert.ToInt32(this.lblLnk_two.Text);

    this.DataHal(this.now);

    this.CekLink();
}

private void Tampil(int x, int y)
{
    int i = 1;

    string idx = "", gb = "gb_tmpt";

    foreach (Control ctrl in this.Controls)
    {
        if (ctrl is GroupBox)
        {
            this.tmpt = new Tempat(1st[x]);
            this.tmpt.SetDataDB();

            if (i != 1)
                idx = i.ToString();

            if (ctrl.Name == gb + idx)
            {
                if (!ctrl.Visible)
                    ctrl.Visible = true;

                foreach (Control ctrl2 in ctrl.Controls)
                {
                    if (ctrl2 is PictureBox)
                    {
                        PictureBox pic = ctrl2 as PictureBox;

                        pic.ImageLocation = tmpt.src;
                        pic.SizeMode = PictureBoxSizeMode.StretchImage;
                    }
                    elseif (ctrl2 is Label)
                    {
                        if (ctrl2.Name == "lbl_isi_nama" + idx)
                            ctrl2.Text = tmpt.nama;
                        elseif (ctrl2.Name == "lbl_isi_almt" + idx)
                            ctrl2.Text = tmpt.almt;
                    }
                    elseif (ctrl2 is Button)
                    {
                        this.RemoveClickEvent(ctrl2 as Button);
                        ctrl2.Click +=
                            new EventHandler(this.LhtRute);
                    }
                }
            }

            i++;
        }
    }
}

```

```

        x++;
    }

    if (i > 12 || x >= y || x == lst.Count())
    break;
    }
}

private void lbllnk_tri_LinkClicked(object sender,
LinkLabelLinkClickedEventArgs e)
{
    this.now = Convert.ToInt32(this.lbllnk_tri.Text);

    this.DataHal(this.now);

    this.CekLink();
}

private void CekLink()
{
    if (Convert.ToInt32(this.now) == this.last)
    this.lbllnk_next.Enabled = false;
    else
    this.lbllnk_next.Enabled = true;

    if (Convert.ToInt32(this.now) == this.fst)
    this.lbllnk_prev.Enabled = false;
    else
    this.lbllnk_prev.Enabled = true;
}

private void UpdLink(int type)
{
    int x;

    if (type == 1)
    {
        if (Convert.ToInt32(this.lbllnk_1st.Text) + 4 > this.last)
        {
            if (Convert.ToInt32(this.lbllnk_1st.Text) + 3 > this.last)
            {
                if (Convert.ToInt32(this.lbllnk_1st.Text) + 2 > this.last)
                {
                    if (Convert.ToInt32(this.lbllnk_1st.Text) + 1 > this.last)
                    x = 0;
                }
            }
        }
        x = 1;
    }
    else
        x = 2;
    }
    else
        x = 3;
    }
    else
        x = 4;
}

```

```

this.lbl1nk_fst.Text = (Convert.ToInt32(this.lbl1nk_fst.Text) +
x).ToString();
this.lbl1nk_two.Text = (Convert.ToInt32(this.lbl1nk_two.Text) +
x).ToString();
this.lbl1nk_tri.Text = (Convert.ToInt32(this.lbl1nk_tri.Text) +
x).ToString();
this.lbl1nk_bfr1st.Text = (Convert.ToInt32(this.lbl1nk_bfr1st.Text) +
x).ToString();
this.lbl1nk_1st.Text = (Convert.ToInt32(this.lbl1nk_1st.Text) +
x).ToString();
    }
else
    {
if (Convert.ToInt32(this.lbl1nk_fst.Text) - 4 < this.fst)
    {
if (Convert.ToInt32(this.lbl1nk_fst.Text) - 3 < this.fst)
    {
if (Convert.ToInt32(this.lbl1nk_fst.Text) - 2 < this.fst)
    {
if (Convert.ToInt32(this.lbl1nk_fst.Text) - 1 < this.fst)
        x = 0;
else
        x = 1;
    }
else
        x = 2;
    }
else
        x = 3;
    }
else
        x = 4;

this.lbl1nk_fst.Text = (Convert.ToInt32(this.lbl1nk_fst.Text) -
x).ToString();
this.lbl1nk_two.Text = (Convert.ToInt32(this.lbl1nk_two.Text) -
x).ToString();
this.lbl1nk_tri.Text = (Convert.ToInt32(this.lbl1nk_tri.Text) -
x).ToString();
this.lbl1nk_bfr1st.Text = (Convert.ToInt32(this.lbl1nk_bfr1st.Text) -
x).ToString();
this.lbl1nk_1st.Text = (Convert.ToInt32(this.lbl1nk_1st.Text) -
x).ToString();
    }
}

private void LhtRute(object sender, EventArgs e)
{
Button btn = (Button)sender;
string idx = "";
int x = this.now;

for (int i = 0; i < gidx; i++)
{
if (i != 0)
idx = ( i + 1 ).ToString();

```

```

if (btn.Name == "btn_rute" + idx)
{
    x = (this.now-1) * 12 + i;

    this.tmpt = newTempat(this.lst[x]);
    this.tmpt.SetDataDB();

    break;
}

RuteFrmrte = newRuteFrm(Convert.ToSingle(this.tmpt.lat),
Convert.ToSingle(this.tmpt.lng));
rte.ShowDialog();
}
privatevoidRemoveClickEvent(Button b)
{
    FieldInfo f1 = typeof(Control).GetField("EventClick", BindingFlags.Static |
BindingFlags.NonPublic);
    object obj = f1.GetValue(b);
    PropertyInfo pi = b.GetType().GetProperty("Events", BindingFlags.NonPublic |
BindingFlags.Instance);
    EventHandlerList list = (EventHandlerList)pi.GetValue(b, null);
    list.RemoveHandler(obj, list[obj]);
}
}
}

using System;
usingSystem.Collections.Generic;
usingSystem.ComponentModel;
usingSystem.Data;
usingSystem.Drawing;
usingSystem.Linq;
usingSystem.Text;
usingSystem.Threading.Tasks;
usingSystem.Windows.Forms;
usingMicrosoft.Office.Interop;

namespacePlace_Recognition
{
    publicpartialclassUjiFrm : Form
    {
        privateclsfunctionclsfunc;

        publicUjiFrm()
        {
            InitializeComponent();
            this.WindowState = FormWindowState.Maximized;

            this.clsfunc = newclsfunction();

            List<string>lst = this.clsfunc.GetAllUji();

            for (inti = 0; i<lst.Count(); i++)
            {
                Pengujian uji = newPengujian(lst[i]);
            }
        }
    }
}

```

```

        uji.SetDataDB2();

        Tempat tmp = new Tempat(uji.limg1);
        tmp.SetDataDB();

        this.dg_uji.Rows.Add(i + 1, uji.limg2, tmp.citra.link, tmp.nama, tmp.almt,
            uji.precision, uji.recall, uji.time);
    }
}

private void copyAlltoClipboard()
{
    this.dg_uji.ClipboardCopyMode =
        DataGridViewClipboardCopyMode.EnableAlwaysIncludeHeaderText;
    this.dg_uji.MultiSelect = true;
    this.dg_uji.SelectAll();
    DataObject dataObj = this.dg_uji.GetClipboardContent();
    if (dataObj != null)
        Clipboard.SetDataObject(dataObj);
}

private void Button1_Click(object sender, EventArgs e)
{
    DialogResult x1 = MessageBox.Show("Download Excel?", "Perhatian !!!",
        MessageBoxButtons.YesNo);

    if (x1 == DialogResult.Yes)
    {
        copyAlltoClipboard();
        Microsoft.Office.Interop.Excel.Application xlexcel;
        Microsoft.Office.Interop.Excel.Workbook xlWorkBook;
        Microsoft.Office.Interop.Excel.Worksheet xlWorkSheet;
        object misValue = System.Reflection.Missing.Value;
        xlexcel = new Microsoft.Office.Interop.Excel.Application();
        xlexcel.Visible = true;
        xlWorkBook = xlexcel.Workbooks.Add(misValue);
        xlWorkSheet =
            (Microsoft.Office.Interop.Excel.Worksheet)xlWorkBook.Worksheets.get_Item(1);
        Microsoft.Office.Interop.Excel.Range CR =
            (Microsoft.Office.Interop.Excel.Range)xlWorkSheet.Cells[1, 1];
        CR.Select();
        xlWorkSheet.PasteSpecial(CR, Type.Missing, Type.Missing, Type.Missing,
            Type.Missing, Type.Missing, true);
    }
}

}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

```

```

namespace Place_Recognition
{
    public partial class TentangFrm : Form
    {
        public TentangFrm()
        {
            InitializeComponent();
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Place_Recognition
{
    public partial class ProcessFrm : Form
    {
        public Action worker;
        public ProcessFrm(Action worker)
        {
            InitializeComponent();

            if (worker == null)
                throw new ArgumentNullException();

            this.worker = worker;
        }

        protected override void OnLoad(EventArgs e)
        {
            base.OnLoad(e);
            Task.Factory.StartNew(this.worker).ContinueWith(t => { this.Close(); },
                TaskScheduler.FromCurrentSynchronizationContext());
        }
    }
}

using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Threading;
using System.Windows.Forms;
using System.Device.Location;
using System.Net;

```

```

using System.IO;
using GMap.NET;
using GMap.NET.CacheProviders;
using GMap.NET.Internals;
using GMap.NET.ObjectModel;
using GMap.NET.Projections;
using GMap.NET.WindowsForms;
using GMap.NET.WindowsForms.Markers;
using GMap.NET.MapProviders;

namespace Place_Recognition
{
    public partial class RuteFrm : Form
    {
        double nlat, nlng, clat, clng;

        clsfunction clsfunc;
        public RuteFrm(float lat, float lng)
        {
            InitializeComponent();
            this.WindowState = FormWindowState.Maximized;

            this.clsfunc = new clsfunction();
            this.clat = Convert.ToDouble(lat);
            this.clng = Convert.ToDouble(lng);
            this.nlat = Convert.ToDouble(lat);
            this.nlng = Convert.ToDouble(lng);
        }
        public bool CheckCon()
        {
            try
            {
                {
                    using (var client = new WebClient())
                    using (client.OpenRead("http://clients3.google.com/generate_204"))
                    {
                        return true;
                    }
                }
            }
            catch
            {
                return false;
            }
        }

        private void RuteFrm_Load(object sender, EventArgs e)
        {
            PointLatLng start = new PointLatLng(this.clat, this.clng);

            this.gmap.MapProvider = GMapProviders.GoogleMap;
            this.gmap.MinZoom = 0;
            this.gmap.MaxZoom = 24;
            this.gmap.Zoom = 18;
            this.gmap.Position = start;
            this.gmap.CanDragMap = true;
            this.gmap.DragButton = MouseButtons.Left;
            this.gmap.ShowCenter = false;
            GMapProviders.GoogleMap.ApiKey = clsfunc.GetGoogleApi();
        }
    }
}






```

```
GMapMarker marker = newGMarkerGoogle(start, GMarkerGoogleType.red);
GMapOverlay markers = newGMapOverlay("markers");
markers.Markers.Add(marker);
this.gmap.Overlays.Add(markers);
    }
}
```


KARTU BIMBINGAN SKRIPSI

KARTU BIMBINGAN SKRIPSI	
Semester Gasal/Genap Tahun Akademik 2019/2020	
Nama : May Hendra Putra Sit	Pembimbing I : Dr. Mhd Arqam, S.S., M.Comp.Sc
NIM : 0701163068	Pembimbing II : Rethmat Kurniawan R. S.T.M.Kom
Prog. Studi : Ilmu komputer	SK Pembimbing :
Judul Skripsi : <u>Penerapan Metode Speed Up Robust Features (SURF) dan Features from Accelerated Segment Test (FAST) untuk pengendalian tempur.</u>	

P E R T	PEMBIMBING I			PEMBIMBING II		
	Tgl.	Materi Bimbingan	Tanda Tangan	Tgl.	Materi Bimbingan	Tanda Tangan
I	8 mei 2020	BAB 2	<i>Fugan</i>	17 maret 2020	BAB I	<i>M</i>
II	12 mei 2020	BAB 3	<i>Fugan</i>	21 maret 2020	BAB II	<i>M</i>
III	17 mei 2020	Pengecepan Proposal Skripsi	<i>Fugan</i>	7 April 2020	BAB III	<i>M</i>
IV	18 mei 2020	Acc Proposal skripsi	<i>Fugan</i>	13 April 2020	Pengecepan Proposal Skripsi	<i>M</i>
V				6 mei 2020	Acc Proposal skripsi	<i>M</i>

VI	20 okt 2020	Revisi Bab IV	Fugan.	31 Agus 2020	Revisi Proposal	
VII	21 okt 2020	Revisi Bab V	Fugan.	7 okt 2020	Revisi Bab IV & V	
VIII	22 okt 2020	Revisi Abstrak, kesimpulan, Daftar pustaka	Fugan.	17 okt 2020	Revisi Abstrak, kesimpulan, Daftar pustaka	
IX	23 okt 2020	Acc Semua Bab	Fugan.	20 okt 2020	Acc Semua Bab	
X	23 okt 2020	Acc sidang	Fugan.	23 okt 2020	Acc Sidang	

Medan, 23 Oktober 2020
An. Dekan
Ketua Jurusan/Program Studi



Dr. Mhd Fugan, S.Si., M. Comp. Sc
NIP. 198008062006041003

Catatan: Pada saat bimbingan, kartu ini harus diisi dan ditandatangani oleh pembimbing

DAFTAR RIWAYAT HIDUP



Data Diri

Nama : Mey Hendra Putra Sirait
Nim : 0701163068
Tempat & Tanggal Lahir : Tanjungbalai, 21 Mei 1998
Jenis Kelamin : Laki-Laki
Alamat : Jalan Ustad Abdul Hamid
Kecamatan/Kabupaten : Tanjungbalai Selatan
Agama : Islam
No Hp : 082276260180
Email : Meyhendraputra@gmail.com

Nama Orang tua

Ayah : Ramlan Sirait
Ibu : Ramsiah Siregar

Pendidikan Formal

1. SD Negeri 5 (13204) Tanjungbalai (2004-2010)
2. SMP Negeri 1 Tanjungbalai (2010-2013)
3. SMA Negeri 1 Tanjungbalai (2013-2016)